МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ АВІАЦІЙНИЙ УНІВЕРСИТЕТ
Факультет кібербезпеки, комп’ютерної та програмної інженерії
Кафедра комп’ютерних систем та мереж

“ДОПУСТИТИ ДО ЗАХИСТУ”
Завідувач кафедри
__________________ Жуков І.А.
“_____”___________2020 р.

ДИПЛОМНА РОБОТА

(ПОЯСНЮВАЛЬНА ЗАПИСКА)

випускника освітнього ступеня “МАГІСТР”

спеціальності 123 «Комп’ютерна інженерія»

освітньо-професійної програми «Комп’ютерні системи та мережі»

на тему: **“Веб-застосунок з організації та адміністрування мовних курсів”**

Виконавець: ____________________________________ Чернецький В.В.

Керівник: _____________________________________ Іскренко Ю.Ю.

Нормоконтролер: ______________________________ Надточій В.І.

Засвідчую, що у дипломній роботі
немає запозичень з праць інших авторів
без відповідних посилань
Чернецький В.В.

Київ 2020

MINISTRY OF EDUCATION AND SCIENCE OF UKRAINE
NATIONAL AVIATION UNIVERSITY
Faculty of Cybersecurity, Computer and Software Engineering
Computer Systems and Networks Department

“PERMISSION TO DEFEND GRANTED”

The Head of the Department

___________________ Zhukov I.A.

“_____”____________2020

MASTER’S DEGREE THESIS

(EXPLANATORY NOTE)

Specialty: 123 Computer Engineering

Educational-Professional Program: Computer Systems and Networks

Topic: **“Web application for organizing and administrating of language courses”**

Completed by: ____________________________________Chernetskyi V.V.

Supervisor: _______________________________________ Iskrenko Yu.Yu.

Standards Inspector: ________________________________ Nadtochii V.I.

Kyiv 2020

НАЦІОНАЛЬНИЙ АВІАЦІЙНИЙ УНІВЕРСИТЕТ

Факультет кібербезпеки, комп’ютерної та програмної інженерії
Кафедра комп’ютерних систем та мереж
Освітній ступінь: «Магістр»
Спеціальність: 123 «Комп’ютерна інженерія»
Освітньо-професійна програма: «Комп’ютерні системи та мережі»

“ЗАТВЕРДЖУЮ”
Завідувач кафедри
__________________ Жуков І.А.
“_____”________________2020 р.

ЗАВДАННЯ
на виконання дипломної роботи
Чернецького Віталія Владиславовича
(прізвище, ім'я та по-батькові випускника в родовому відмінку)

1. Тема дипломної роботи: “Веб-застосунок з організації та адміністрування мовних курсів” затверджена наказом ректора від 25.09.2020 р. № 1793/ст

2. Термін виконання роботи (проекту): з 1 жовтня 2020 р. до 25 грудня 2020 р.

3. Вихідні дані до роботи (проекту): *Програмні засоби для розробки програмного забезпечення та створення баз даних.*

4. Зміст пояснювальної записки: *Вступ, огляд теми, огляд існуючих технологій побудови веб-застосунків, баз даних, мов програмування, патернів програмування, фреймворків, розробка веб-застосунку, пояснення роботи веб-застосунку, висновки по роботі.*

5. Перелік обов'язкового графічного (ілюстративного) матеріалу: *Графічні матеріали результатів дослідження надати у вигляді презентації у форматах .ppt, .pdf.*

NATIONAL AVIATION UNIVERSITY

Faculty of Cybersecurity, Computer and Software Engineering
Department: Computer Systems and Networks
Educational Degree: “Master”
Specialty: 123 “Computer Engineering”
Educational-Professional Program: “Computer Systems and Networks”

“APPROVED BY”
The Head of the Department
__________________Zhukov I.A.
“_____”_______________2020 p.

Graduate Student’s Degree Thesis Assignment

Chernetskyi Vitalii Vladyslavovych

1. Thesis topic: “Web application for organizing and administrating of language courses” approved by the Rector’s order of 25.09.2020 р. № 1793/ст

2. Thesis to be completed between 01.10.2020 and 25.12.2020.

3. Initial data for the project (thesis): *Software tools for software developing and databases creating.*

4. The content of the explanatory note: *Introduction, overview of the topic, overview of the existing technologies for web application and databases designing, programming languages, software design patterns, frameworks, web application developing, explanation of web application running, conclusions on work.*

5. The list of mandatory graphic materials: *Graphic materials are given in MS Power Point presentation.*

6. Календарний план-графік

№ пор.	Завдання	Термін Виконання	Підпис керівника
1	Узгодити технічне завдання з керівником дипломної роботи	1.10.20- 8.10.20	
2	Виконати пошук та вивчення науково-технічної літератури за темою роботи	9.10.20- 15.10.20	
3	Опрацювати теоретичний матеріал	16.10.20- 18.10.20	
4	Огляд існуючих технологій для розробки веб-застосунку	19.10.20- 03.11.20	
5	Розробка та тестування веб-застосунку	04.11.20- 15.12.20	
6	Оформлення пояснювальної записки	06.12.20- 12.12.20	
7	Оформити графічну частину записки та подати матеріали работи на антиплагіатну перевірку матеріалів	13.12.20- 14.12.20	
8	Отримати рецензію та відгук керівника. Надати матеріали роботи на кафедру.	15.12.20 18.12.20	

7. Дата видачі завдання: “1” жовтня 2020 р.

Керівник дипломної роботи ________________________ Іскренко Ю.Ю.
(підпис керівника)

Завдання прийняв до виконання ____________________ Чернецький В.В.
(підпис випускника)

6. TIMETABLE

#	Completion stages of Degree Project (Thesis)	Stage Completion Dates	Signature of the supervisor
1	Technical task coordination with supervisor	1.10.20-8.10.20	
2	Research and study scientific and technical literature according to the theme	9.10.20-15.10.20	
3	Working on theoretical information	16.10.20-18.10.20	
4	Overviewing of existing technologies for web application development	19.10.20-03.11.20	
5	Web application developing and testing	04.11.20-15.12.20	
6	Explanatory note preparing	06.12.20-12.12.20	
7	Preparation of the graphical part and sending materials of work to antiplagiarism checking	13.12.20-14.12.20	
8	Receiving the reviews from reviewer and supervisor and providing the materials to the department	15.12.20 18.12.20	

7. Assignment issue date: 1.10.2020

Diploma Thesis Supervisor ______________________________Iskrenko Yu.Yu.
(Signature)

Assignment accepted for completion ________________________Chernetskyi V.V.
(Student’s Signature)

ABSTRACT

The Explanatory Note on Master's Degree Graduation Project – "Web application for organizing and administrating of language courses": 89 pages, 60 figures, 28 references.

WEB APPLICATION, USER'S CONTROL, BACKEND, FRONTEND, FRAMEWORKS, PROGRAMMING, DATABASE, DEVELOPING.

The Goal of Graduation Project: The goal of this project is to develop web application that will have possibility for admin user to perform some manipulation on pages.

Main Tasks: Technologies observing, suitable programming patterns implementing, programming language and frameworks choosing, web application developing.

The Subject of Project: Define most suitable technologies, programming languages, frameworks, programming patterns; combine them into one system with help of appropriate software.

Practical usage: Creating platform for language school to represent their language courses. Possibility to manage website provides lower cost support spending.

Main Metrics and Results: With today's high level of rivalry between private schools, it is necessary for them to have portal where they can represent own courses and to have ability manage it without additional payments to developers. Web application provides such possibilities.

CONTENTS

LIST OF ABBREVIATIONS

OOP	- Object Oriented Programming
HTML	- Hypertext Markup Language
UI	- User Interface
VM	- Virtual Machine
POP3	- Post Office Protocol version 3
IMAP	- Internet Message Access Protocol
CSS	- Cascading Style Sheets
DOM	- Document Object Model
CPU	- Central Processing Unit
AJAX	- Asynchronous JavaScript And XML
CRUD	- Create Read Update Delete
SDK	- Software Development Kit
MVC	- Model View Controller
DB	- Database
W3C	- World Wide Web Consortium
ACID	- Atomicity, Consistency, Isolation, Durability
SQL	- Structured Query Language
RDBMS	- Relational Database Management System
IDE	- Integrated Development Environment
JDK	- Java Development Kit
JRE	- Java Runtime Environment
JDBC	- Java Database Connectivity
BLOB	- Binary Large Object
DAO	- Data Access Object
JPA	- Java Persistence API

INTRODUCTION

Simple websites does not provide much of functionality. Different companies, including language schools, need tool to publish their propositions. It is important for them to have possibility to rule web pages layout without additional payments for developers.

Modern web development concentrates not only on static or dynamic websites. Web application is a program user interface of which is displayed as website. It provides possibilities for data processing.

The goal of this work is to create application that will provide possibilities for user to control pages layout. At the same time, it should store changes in database. Web application should perform some checking to prevent user to upload unnecessary data.

Created web application allows user to manipulate slider layout on main page, control blocks with workers description and to form list of courses available. Web application prevent user to upload unnecessary data.

To develop this application, research of technical reports on the Internet was made. New technologies appear much faster than new books can be published. Instead of paper literature, online websites of big variety of scientific and technical organizations publish their researches online.

Developed web application executes all its data processing and data layout view on server, which simplifies usage of it for user. Due to such decision, less data are transferring over the Internet that is better for user with slow Internet connection. At the same time, user browser have not to process data so that it performs quicker on slow devices. At the same time, pages can dynamically suit for small and big screens.

PART 1
OVERVIEW OF SOME OF EXISTING TECHNOLOGIES AND METHODS OF WEB-APPLICATION DEVELOPMENT

1.1. Most Used Programming Languages

For creating any program, it is used programming language. In the time of first computer appearance, programs were written on machine language. Machine language is a low-level language comprised of binary digits. Nowadays most developers don't use machine language to develop programs. They use languages that are more understandable for humans. The most used programming languages as for February 2020 in Ukraine are shown below (fig. 1.1) [1].

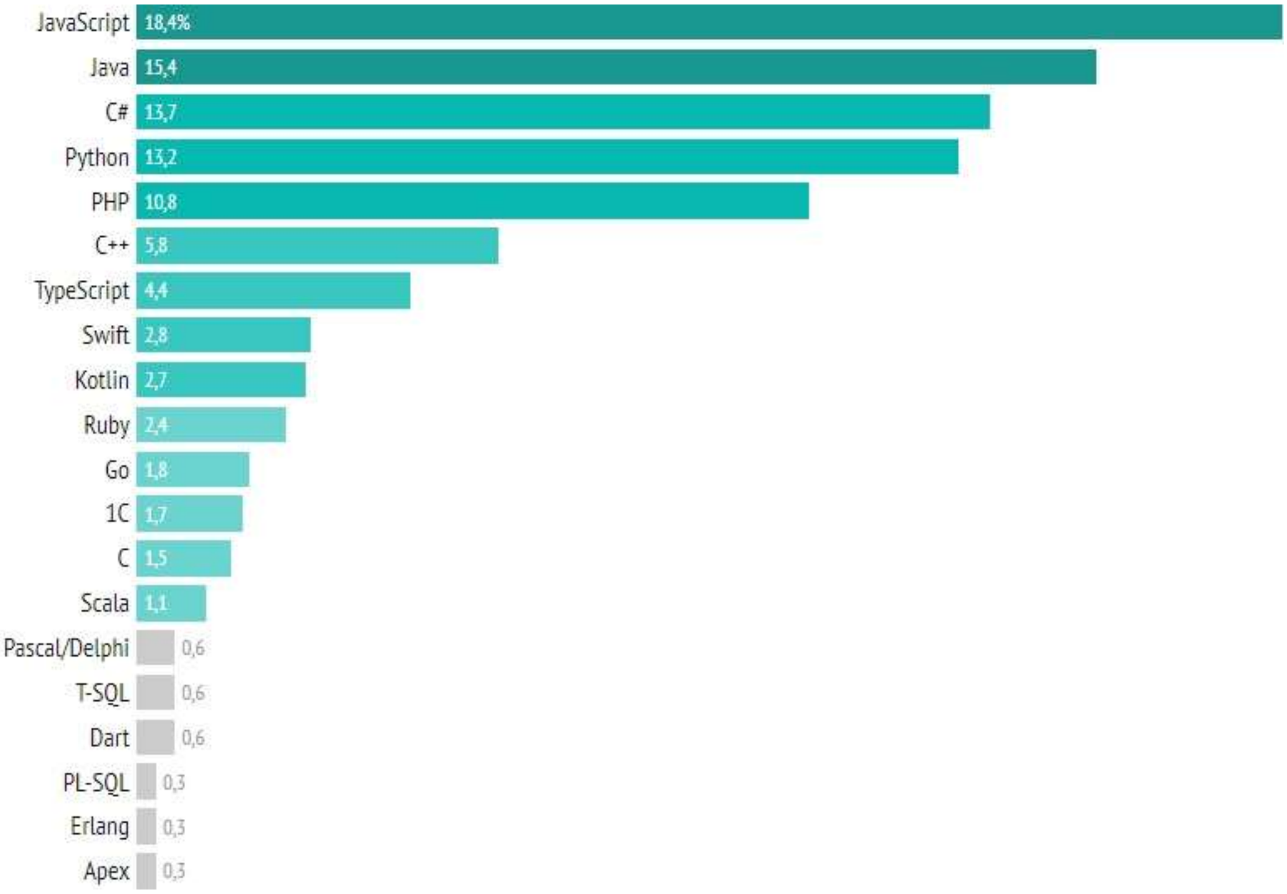


Fig. 1.1. Most used languages in commercial projects in Ukraine in 2020

1.1.1. JavaScript

JavaScript is a dynamic computer programming language. It is high-level and most commonly used for creating dynamic web pages. It is has OOP possibilities.

Client-side JavaScript is the most used form of the language. The script should be included in or referenced by an HTML document for the code to be interpreted by the browser. It means that a web page need not be a static HTML, but can include programs that interact with the user, control the browser, and dynamically create HTML content. It is possible use JavaScript to check if the user has typed a valid e-mail address in a form field. The JavaScript code is performed when the user press submit button in the form, and only in the case of all inputs are valid, they would be passed to the Web Server. JavaScript can be used to catch user-performed events such as button clicks, link navigation, and other actions that the user initiates explicitly or implicitly.

Advantages of JavaScript:

1) less server interaction – it is possible to check user input before sending the data to the server. This saves server traffic, which means less load on the server;

2) immediate feedback to the visitors − users do not have to wait until a page reloaded to see if they have forgotten to input something;

3) increased interactivity – it is possible to make UI that react when the user move over them with a mouse or activates them via the keyboard;

4) richer interfaces – it is possible to use JavaScript to include such items as drag-and-drop components and sliders to give a Rich Interface to the site visitors.

It is not possible to use JavaScript as a full-performed programming language. It has such disadvantages:

1) client-side JavaScript does not able to read and write files. Such situation is due to security reasons;

2) JavaScript does not support networking applications;

3) JavaScript doesn’t able to perform multithreading activity [2].

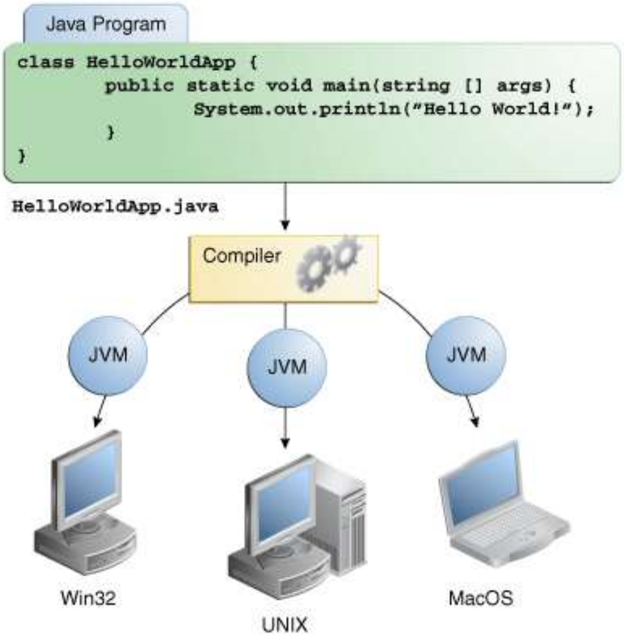
1.1.2. Java

Java is a high-level, cross-platform OOP language that was created by Sun Microsystems in the year 1995. Nowadays, Java is used to run different apps such as games, social media apps, audio and video applications, etc. It is mainly used for enterprise programs. This language can characterized with such words:

- Simple
- Object oriented
- Distributed
- Multithreaded
- Dynamic
- Architectural neutral
- Portable
- High performance
- Robust
- Secure

Simple means that language syntax and structure is easy to learn. This language can be easily understand for beginners. Object-oriented means that Java is developed with OOP in mind. At the same time, it is not functional language. Java is based on a biological theory of the origin of languages, which means that Java is distributed language. In addition, this language support multithreading. It means that it is possible to create parallel programs that executed in parallel threads. In case of using multicore processors, parallel programs will be executed faster comparing with one thread program. Dynamic language means that it at runtime execute different programming behaviors that static language executes during compilation. Such feature allows adding new code, extending objects and definitions or modifying type system during runtime.

Because the Java VM is available on many different operating systems, the same .class files are capable of running on Microsoft Windows, the Solaris™ Operating System (Solaris OS), Linux, or Mac OS. Some virtual machines, such as the Java SE HotSpot at a Glance, perform additional steps at runtime to give your application a performance boost. This includes various tasks such as finding performance bottlenecks and recompiling (to native code) frequently used sections of code (fig. 1.2).



```
class HelloWorldApp {
        public static void main(string [] args) {
                System.out.println("Hello World!");
        }
}
```

Fig. 1.2. Through the Java VM, the same application is capable of running on multiple platforms.

As a platform-independent environment, the Java platform can be a bit slower than native code. However, advances in compiler and virtual machine technologies are bringing performance close to that of native code without threatening portability [3].

1.1.3. C#

C# is a modern, general-purpose programming language that is used to perform a wide range of tasks and objectives that span over a variety of professions. C# is primarily used on the Windows .NET framework, although it can be applied to an open source platform. This highly versatile programming language is an object-oriented programming language.

Like other general-purpose programming languages, C# is used to make a number of different programs and apps: mobile apps, desktop apps, cloud-based services, websites, enterprise software and games. While C# is remarkably versatile, there are three areas in which it is most commonly used: website development, windows applications, games.

C# also takes simplicity and efficiency in mind, so that programmers need less time to write complicated blocks of code that are used several times throughout the project. With an extensive memory bank and it’s got a time-effective language that can easily reduce many hours and help to meet deadlines [4].

1.1.4. Python

Python is a widely-used, interpreted, object-oriented, and high-level programming language with dynamic semantics, used for general-purpose programming.

Python goals:

- an easy and intuitively understandable language which is powerful at the same time;
- open source, so anybody has possibility to take part in its development;
- code as understandable as simple English;
- suitable for everyday tasks and for short development times.

Python provides different modules to support graphical interface. This language is portable which means that it is not necessary to change code to execute it on different systems such as Linux, Unix, Mac and Windows. It can be integrated with other languages like C++, C, etc. Python executed line per line which makes easier to debug. In addition, Python has large standard library.

1.1.5. PHP

PHP is a server side scripting language that is integrated with HTML. It is used to create dynamic page content. It can process data received from client and give back results of processing. PHP is integrated with all popular databases such as MySQL, PstgreSQL, Oracle, Microsoft SQL Server. It supports many different protocols, including POP3 and IMAP. PHP has C-like syntax.

PHP performs system functions, i.e. from files on a system it can create, open, read, write, and close them. PHP can handle forms: take data from files, store data to a file, work with emails, return data to the user. It is possible to add, delete, modify elements within database using PHP. It has possibilities to access cookies and set cookies. In addition, it is also possible to restrict user access to some pages of website and to encrypt data.

1.2. Most Popular Frameworks, Libraries and Platforms

To make developers' life easier, many different add-ons exist. During development process, large amount of actions are repetitive. In the same time, it is irrational time spending to write the same functionality in every project from a glance. So that, developers use different frameworks, libraries and platforms to speed up development process. The most popular of them are shown below (fig. 1.3) [5].

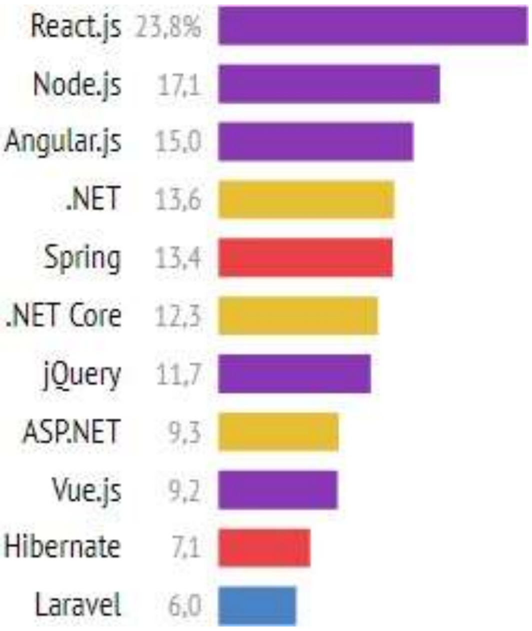


Fig. 1.3. The most popular frameworks, libraries and platforms in Ukraine in 2020

1.2.1. React.js Library

React is a library developed by Facebook and used for frontend development. It is used for handling the interface layer of websites and mobile applications. ReactJS gives possibility to make reusable user interface elements. It is nowadays one of the most used JavaScript libraries and has a big foundation and enormous community participating it. To work with ReactJS, it is necessary to have a good knowledge of JavaScript, HTML5, and CSS.

React is a library for creating composable user interfaces. It allows the creation of reusable UI elements, which displays data that can be changeable. Many people use React as the view component in model-view-controller pattern. React abstracts away the DOM, proposing an easier developing and better performance. React can also execute on the server side using Node, and it can power native applications using React Native. React provides one-way reactive data flow, which reduces the boilerplate and is simpler to reason about than usual data binding [6].

React features:

- JSX – JSX is JavaScript syntax extension. It is recommended to use JSX while developing with React, however, it is not obvious.
- Components – it is central feature of React. It is necessary to have in mind everything as a component. Due to this feature it will be helpful to scale projects.
- Unidirectional data flow and Flux – React provides one-way data flow which makes it simpler to reason about the application. Flux is a pattern for keeping data unidirectional.

React advantages:

- Uses virtual DOM which is a JavaScript object. It improves application productivity, because JavaScript virtual DOM is much faster that the usual DOM.
- React can be executed on client and server side as well as with other frameworks.
- Component and patterns increase readability, that helps to develop larger applications.

React limitations:

- Represents only the view layer of the application, so that it is necessary to use other technologies to get a full tooling set for developing.
- React uses inline templating and JSX, which can be awkward in development.

1.2.2. Node.js Platform

Node.js is a server-side platform built on Google Chrome's JavaScript Engine for easily building fast and scalable network applications. Node.js uses an event-driven, non-blocking I/O model that makes it lightweight and efficient, perfect for data-intensive real-time applications that run across distributed devices. Node.js also provides a rich library of various JavaScript modules which simplifies the development of web applications using Node.js to a great extent. Node.js = Runtime Environment + JavaScript Library.

Features of Node.js:

- Asynchronous and Event Driven − Every API of Node.js library is asynchronous, non-blocking. It fundamentally means a Node.js based server never waits for an API to return data. It moves to the next API after calling it and a notification mechanism of Events of Node.js helps the server to get a response from the previous API call.
- Very Fast – Because it was built on Google Chrome's V8 JavaScript Engine, Node.js library is very fast in code execution.
- Single Threaded but Highly Scalable − Node.js uses a single threaded model with event looping. Event mechanism makes the server to give respond in a non-blocking way and makes the server highly scalable, which is not available for usual servers that create limited threads to handle requests. Node.js uses a single threaded program and the same program can provide service to a much larger number of requests than traditional servers like Apache HTTP Server.

- No Buffering − Node.js apps never buffer any data. These apps simply output the data in chunks.

Areas where it Node.js is perfect technology to use:

1) I/O bound Apps;

2) Data Streaming Apps;

3) Data Intensive Real-time Apps (DIRT);

4) JSON APIs based Apps;

5) Single Page Apps.

Node.js is not recommended to use for CPU intensive applications [7].

1.2.3. AngularJS Framework

AngularJS is a structural framework for dynamic web apps. It provides possibility to use HTML as template language and lets to extend HTML's syntax to display application's elements clearly and succinctly. AngularJS's data binding and dependency injection makes less amount of code that would be written in case of not usage of AngularJS. AngularJS runs on client side so that it is good partner with any server side tool.

AngularJS was built providing declarative code that is better than imperative when it comes to building user interfaces and connecting software elements together, while imperative code is great for expressing business logic.

AngularJS decouples DOM manipulation from app logic. It significantly improves the testability of the code. This framework provides idea of regarding app testing as equal in importance to app writing. Testing difficulty is significantly affected by the way the code is structured.

AngularJS frees developer from the following pains:

Registering callbacks: Registering callbacks messes code, making it harder to find bugs. Removing usual boilerplate code such as callbacks is a good help . It significantly reduces the quantity of JavaScript code, and it makes it easier to understand what the app does.

Ruling HTML DOM by software: Manipulating HTML DOM is a cornerstone of AJAX apps, but it's lumbering and error-prone. By describing how the user interface should change as app state changes, developer is freed from low-level DOM ruling tasks. Most apps written with AngularJS never have to programmatically manipulate the DOM, although developer can when necessary.

Marshaling data to and from the UI: CRUD operations make up the most of AJAX apps' tasks. The flow of marshaling data from the server to an internal object to an HTML form, making possible for users to change the form, validate the form, display validation errors, return to an internal model, and then back to the server, makes a lot of boilerplate code. AngularJS erase most of this boilerplate, making code that describes the overall flow of the app instead of all of the implementation details.

Writing large amounts of initial code just to get started: Typically it is necessary to write a lot of plumbing just to get a basic "Hello World" AJAX app working. With AngularJS it is possible to bootstrap application simply using services, which are auto-injected into app in a Guice-like dependency-injection style. This allows developer to get started developing features very quickly. In addition, developer gets full control over the initialization process in automated tests [8].

1.2.4. .NET Platform

.NET Core is general purpose development platform created by Microsoft. It works across different platforms and was redesigned in an significant way that creates .NET fast, flexible and modern. This platform is one among the foremost contributions by Microsoft. Developers can now build Android, iOS, Linux, Mac, and Windows apps with .NET, bushed Open Source.

Major features of .Net platform.

It is cross-platform:

1) Apps implemented in .NET Core can be executed independent of the platform target;

2) It currently supports three main operating systems (Windows, Linus, MacOS);

3) The supported OS, CPUs and apps' scenarios will grow over time, supported by Microsoft and individuals.

.NET platform is flexible for deployment. There can be two types of deployments for .NET Core apps - framework-dependent deployment and self-contained deployment. With framework-dependent deployment, application depends on a system-wide version of .NET Core on which in this application and third-party dependencies are installed. With self-contained deployment, the .NET Core version used to build given app is also deployed along with it application and third-party dependencies and can run side-by-side with other versions.

All .NET product scenarios can be executed via command-line. In addition, .NET Core is suitable with .NET Framework, Xamarin and Mono, via the .NET Standard Library. .NET platform is modular. It is released through NuGet in smaller assembly packages. This framework is one large assembly that contains most of the core possibilities. .NET Core is made available as smaller feature-centric packages. This modular approach makes possible for the developers to optimize their application by including just those NuGet packages which they need in their application. The benefits of a smaller application surface area include better security, more less servicing, improved productivity, and decreased costs in a pay-for-what-you-use model.

.NET Core Platform includes such parts:

- .NET Runtime − It provides a type system, assembly loading, a garbage collector, native interop and other basic services.
- Fundamental Libraries − A set of framework libraries, that provide primitive data types, application composition types and fundamental utilities.
- SDK & Compiler − A set of SDK tools and language compilers that make available the base developer experience, available in the .NET Core SDK.
- 'dotnet' application host − It is used to launch .NET Core apps. It selects the runtime and hosts the runtime, provides an assembly loading policy and launches the application. The same host is also used to startup SDK tools in the same way [9].

1.2.5. Spring Framework

Spring is very popular app designing framework for Java enterprise. Millions of programmers in the world use Spring Framework to create high productive, easily testable, and reusable code.

Spring is lightweight when it comes to size and transparency. The basic version of Spring framework is around 2MB.

The major features of the Spring Framework can be used in designing any Java app, but there are extensions for creating web-apps on top of the Java EE platform. Spring framework needs to make J2EE development simpler to use and provides good developing practices by introducing a POJO-based programming model.

There are such benefits that Spring Framework provides:

1) Spring allows developers to design enterprise-class apps using POJOs. The feature of using only POJOs is that it is no need in an EJB container product such as an app server but developer has a possibility of using only a robust servlet container such as Tomcat or some commercial product.

2) Spring is created in a modular form. Even though the quantity of packages and classes are substantial, developer has to keep with only the ones developer needs and ignore the others.

3) Spring does not reinvent the wheel, instead it truly makes use of some of the existing technologies like several ORM frameworks, logging frameworks, JEE, Quartz and JDK timers, and other view technologies.

4) Testing an app written with Spring is easy due to environment-dependent code that is moved into this framework. Moreover, by using JavaBeanstyle POJOs, it becomes simpler to use dependency injection for injecting test data.

5) Spring's web framework is a well-developed web MVC framework, which gives a good alternative to web frameworks such as Struts or other over-engineered or less popular web frameworks.

6) Spring introduces a comfortable API to translate technology-specific exceptions (thrown by JDBC, Hibernate, or JDO, for example) into solid, unchecked exceptions.

7) Lightweight IoC containers are created to be lightweight, especially when compared to EJB containers. This is useful for designing and deploying apps on computers with limited memory and CPU resources.

8) Spring introduces a strong transaction management interface that can scale down to a local transaction (using a single database) and scale up to global transactions (using JTA, for example) [10].

1.3. Software Design Patterns

Software design pattern is a general, reusable solution to a commonly occurring problem within a given context in software design. It is not a finished design that can be transformed directly into source or machine code. Rather, it is a description or template for how to solve a problem that can be used in many different situations. Design patterns are formalized best practices that the programmer can use to solve common problems when designing an application or system.

1.3.1. Singleton Pattern

Singleton is refer to creation of one class object. It is good decision when one (and only one) object is necessary to perform actions across the system. There are several examples of where only a single instance of a class should exist, such as caches, thread pools, registries and so on.

Usually, class that is used as singleton receive private access modifier. This means that only the members of the class has access to the private constructor and no one else.

There are exist possibility to subclass a singleton using protected constructor instead of private. It can be suitable under some circumstances. One reason taken in these options is to make a register of singletons of the subclasses and the getInstance method has possibility to take in a parameter or use an environment variable for returning the necessary singleton. The registry then contains a mapping of string names to singleton objects, which can be used when necessary [11].

Singleton pattern has its own advantages and disadvantages. To advantages belongs such points:

1) Developer can be sure that a class has only one instance.

2) Developer gain a global access point to that single instance.

3) The singleton object is initialized only when it is called for the first time.

To disadvantages belongs such points:

1) This pattern violates the Single Responsibility Principle so that the pattern solves two problems at the time.

2) The Singleton pattern can mask bad design, for instance, when the components of the program know too much about each other.

3) The pattern requires special treatment in a multithreaded environment so that multiple threads will not create a singleton object several times.

4) It may be difficult to unit test the client code of the Singleton because many test frameworks rely on inheritance when producing mock objects. Since the constructor of the singleton class is private and overriding static methods is impossible in most languages, it will need to think of a creative way to mock the singleton. Alternative is just do not write the tests. Or do not use the Singleton pattern [12].

1.3.2. Factory Method

An usual factory produces goods; a software factory produces objects. Moreover — it performs this without specifying the exact class of the object that must be created. To perform it, objects are created by requesting a factory method instead of requesting a constructor.

Casually, object creation (in Java) takes place like this: SomeClass someClassObject = new SomeClass(); The problem with this decision is that the code using the SomeClass’s object, suddenly now becomes dependent on the exact implementation of SomeClass. It is not bad to use new to create objects but it comes with the baggage of some coupling code to the concrete implementation class, which can sometimes be problematic.

Advantages of Factory method:

1) It is possible to avoid tight coupling between the creator and the exact products.

2) Single Responsibility Principle. Developer can move the product creation code into one place in the program, making the code simpler to maintain.

3) Open/Closed Principle. Developer can provide new types of products into the program without breaking client mode that already exist.

Disadvantage of the method:

1) The code can become more sophisticated since it is necessary to provide many new subclasses to implement the pattern. The best-case scenario is when developer provides the pattern into an existing hierarchy of creator classes.

1.3.3. Strategy Pattern

The strategy pattern allows grouping related algorithms under an abstraction, which allows switching out one algorithm or policy for another without modifying the client. Instead of directly implementing a single algorithm, the code receives runtime instructions specifying which of the group of algorithms to run.

Strategy is a behavioral design pattern that provide possibility for developer to define a family of algorithms, put each of them into a separate class, and create their objects interchangeable.

Advantages of strategy pattern:

1) Developer can exchange algorithms that were used inside an object while runtime.

2) Developer has possibility to isolate the implementation details of an algorithm from the code that uses it.

3) Programmer has possibility to exchange inheritance with composition.

4) Open/Closed Principle. Developer has possibility to introduce new strategies without having to change the context.

Disadvantages of this pattern are:

1) In the case when developer only has a couple of algorithms and they not often changed, there is no worth reason to overcomplicate the code with new classes and interfaces that come along with the pattern.

2) Clients must be aware of the differences between strategies to be able to select a proper one.

3) Many modern programming languages have functional type support that provides possibility for developer to implement different versions of an algorithm inside a set of anonymous functions. Then developer can use such functions exactly as you'd have used the strategy objects, but without bloating your code with extra classes and interfaces [13].

1.3.4. Observer Pattern

Observer pattern is a one-to-many dependency between objects. Due to this when one object swap state, all its dependents are notified. This is usually done by requesting one of their methods.

For example, take in account what happens when user follow someone on Twitter. User are essentially requesting Twitter to send user (the observer) tweet updates of the person (the subject) who was followed. The pattern consists of two sides, the observer who is interested in the updates and the subject who generates the updates.

A subject can have many observers and is a one to many relationship. Nevertheless, an observer is also free to subscribe to updates from other subject. user can subscribe to news feed from a Facebook page, which would be the subject and whenever the page has a new post, the subscriber would see the new post.

Advantages of observer pattern are:

1) Open/Closed Principle. Developer has possibility to provide new subscriber classes without having to swap the publisher's code (and vice versa if there's a publisher interface).

2) Developer has possibility to establish relations between objects at runtime.

Disadvantages of this pattern are: subscribers are notified in random order [14].

1.3.5. MVC Pattern

MVC Pattern stands for Model-View-Controller Pattern. This pattern is used to separate application into independent parts.

Model is an object for carrying data. It can also have logic to update controller if its data changes. View represents the visualization of the data that model contains. Controller acts on both model and view. It controls the data flow into model object and updates the view whenever data changes. It keeps view and model separate (fig. 1.4) [15].

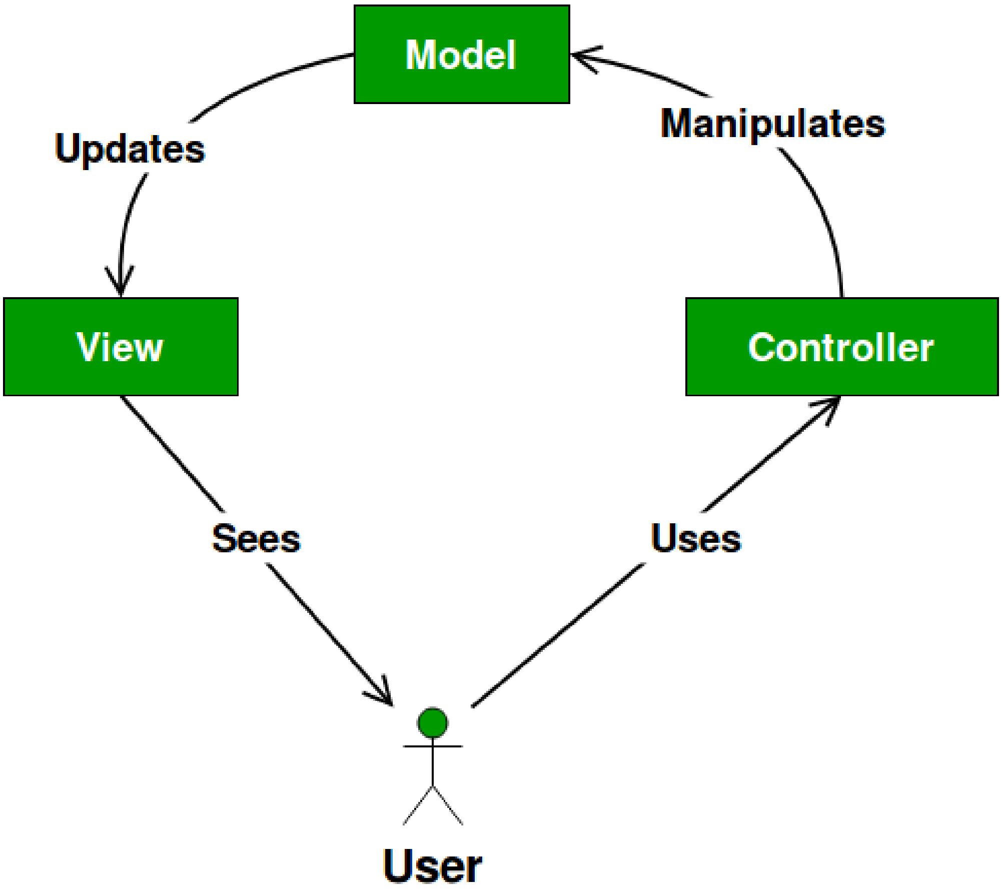


Fig. 1.4. Diagram MVC Design Pattern

MVC is more of an architectural pattern, however, not for full app. MVC mostly relates to the user interface or interaction layer of an app. It is still necessary going to need business logic layer, perhaps some service layer and data access layer.

Advantages of MVC design pattern:

1) Multiple developers are able to work at the same time on the model, controller and views.

2) MVC provides logical grouping of related actions on a controller together. The views for a specific model are also grouped together.

3) Models can have multiple views.

Disadvantages of MVC:

1) The framework navigation can be sophisticated because it provides additional layers of abstraction and requires developers to adapt to the decomposition criteria of MVC [16].

1.4. Databases

A database is a collection of data that is structured so that it can be easily accessed, manipulated and updated. Computer databases usually contain aggregations of data records or files, containing information about sales transactions or interactions with specific customers.

In a relational DB, digital data about a specific customer is structured into rows, columns and tables that are indexed to make it simpler to find necessary data through SQL or NoSQL queries. In contrast, a graph database uses nodes and edges to define relationships between data entries and queries require a special semantic search syntax. As of this writing, SPARQL is the only semantic query language that is approved by the World Wide Web Consortium (W3C).

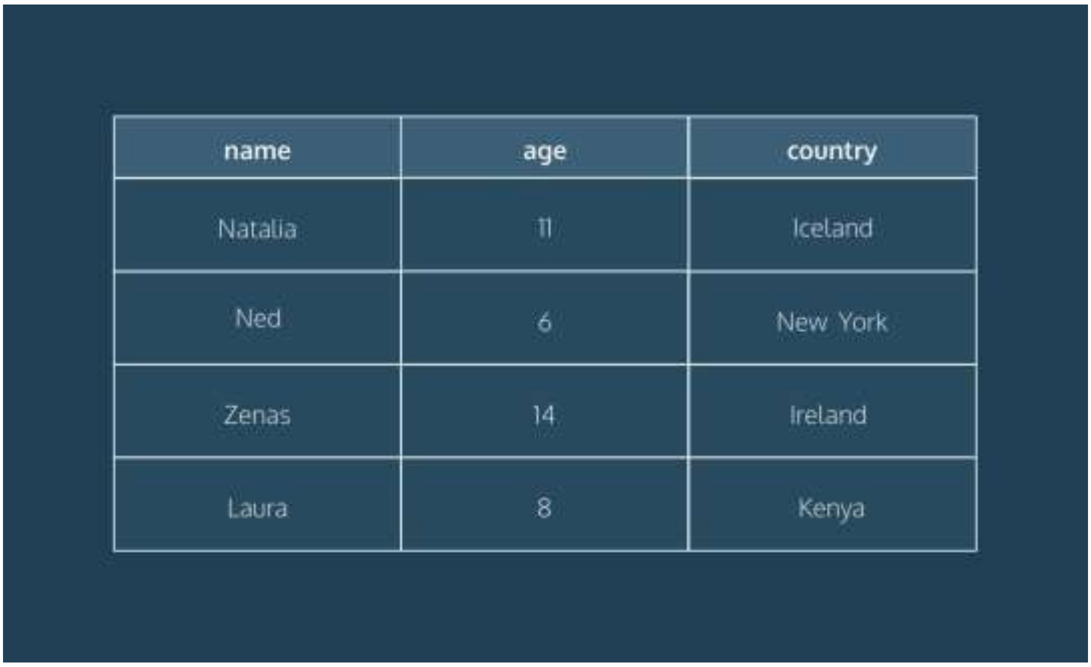
Usually, the DB administrator provides users with the possibility to control read/write access, specify report generation and analyze usage. Some DBs offer ACID (atomicity, consistency, isolation and durability) compliance to guarantee that data is consistent and that transactions are complete [17].

1.4.1. Relational Database

A relational database, invented by E.F. Codd at IBM in 1970, is a tabular DB in which information is specified so that it can be reorganized and accessed in a numerous ways.

Relational DBs are made up of a set of tables with information that fits into a predefined category. Each table has at least one data category in a column, and each row has a certain data instance for the categories which are defined in the columns.

The Structured Query Language (SQL) is the standard user and application program interface for a relational DB. Relational DBs are simple to extend, and a new data category can be added after the original DB creation without requiring modification of all the existing apps. Image representation of the table is shown below (fig. 1.5).



name	age	country
Natalia	11	Iceland
Ned	6	New York
Zenas	14	Ireland
Laura	8	Kenya

Fig. 1.5. Example of table layout in relational DB

A relational database management system (RDBMS) is a program that gives possibility for developer to create, update, and control a relational DB. Most of relational DB management systems use the SQL for accessing the DB.

Many RDBMSs use SQL (and variations of SQL) for accessing the info in tables. For example, SQLite is a relational DB management system. SQLite contains a minimal set of SQL commands (which are the same across all RDBMSs). Other RDBMSs may use other variants.

SQL syntax can be different depending on which RDBMS are used. Here is a brief description of popular RDBMSs:

MySQL. MySQL is the most popular open source SQL DB. It is usually used for web application designing, and usually accessed with PHP. The main advantages of MySQL are that it is simple for use, cheap, reliable (used since 1995), and has a large community of programmers who can help answer questions. Some of the disadvantages are that it is suffer from slow performance while scaling, open source designing has lagged since Oracle has taken control of MySQL, and it is not include some necessary features that programmers may be used to.

PostgreSQL. PostgreSQL is an open source SQL DB that is not belong to any corporation. It is usually used for web application designing. PostgreSQL has many of the same features of MySQL. It is simply to use, cheap, reliable and has an enormous community of programmers. It also implements some other features such as foreign key support without requiring complex configuration. The main disadvantage of PostgreSQL is that it can be slower in performance comparing with other DBs such as MySQL. Its popularity is lower than MySQL.

Oracle Corporation owns Oracle DB, and the code is proprierity. Oracle database is for very big apps, especially in the banking industry. Many of the world's big banks use Oracle apps because Oracle offers a powerful combination of technology and comprehensive, pre-integrated business apps, including fundamental functionality created specifically for banks. The main disadvantage of using Oracle is that it is not free to use like its open source competitors and can be quite expensive.

Microsoft SQL Server. Like Oracle DB, the code proprietary and belongs to Microsoft. Large enterprise apps very often use SQL Server. Microsoft offers a free beginner version called Express but can become very expensive as it is necessary to scale app.

SQLite is a popular open source SQL DB. It can store full DB in one file. One of the most important features this provides is that all of the data can be stored locally without having to connect DB to a server. SQLite is a popular choice for DBs in cellphones, PDAs, MP3 players, set-top boxes, and other electronic gadgets [18].

1.4.2. Distributed Database

A distributed DB is a DB that portions of the DB are stored in multiple physical locations, and in which processing is dispersed or replicated among different points in a network.

Distributed DBs can be homogeneous or heterogeneous. All the physical locations in a homogeneous distributed DB system have the same underlying hardware and run the same OSs and DB apps. The hardware, OSs or DB apps in a heterogeneous distributed DB may be different at each of the locations.

1.4.3. Cloud Database

A cloud DB is a DB that has been optimized or built for a virtualized environment, either in a hybrid cloud, public cloud or private cloud. Cloud DBs provide advantages such as the possibility to pay for storage capacity and bandwidth on a per-use basis, and they provide scalability on demand, along with high availability.

A cloud DB provides enterprises the opportunity to support business apps in a software-as-a-service deployment.

1.4.4. NoSQL Database

NoSQL DBs are useful for large sets of distributed data. NoSQL DBs are effective for Big Data performance issues that relational DBs are not developed to solve. They are the most effective when an organization must analyze large chunks of unstructured data or data that is stored across multiple virtual servers in the cloud.

1.4.5. Object-Oriented Database

Items created using object-oriented programming languages are usually stored in relational DBs, but object-oriented DBs is very suitable for OOP languages.

An object-oriented DB is structured around objects rather than actions, and data rather than logic. For example, a multimedia record in a relational DB can be a definable data object, as instead of an alphanumeric value.

1.4.6. Graph Database

A graph-oriented DB, or graph DB, is a type of NoSQL DB that uses graph theory to store, map and query relationships. Graph DBs are essentially collections of nodes and edges, where each node represents an entity, and each edge represents a connection between nodes.

Graph DBs are becoming more popular for analyzing interconnections. For example, companies may use a graph DB for mining data about their customers from social media profiles.

Graph DBs usually employ SPARQL, a declarative programming language and protocol for graph DB analytics. SPARQL has the possibility to execute all the analytics that SQL can execute, in addition it can be used for semantic analysis, the examination of relationships. It makes it suitable for performing analytics on data sets that have both structured and unstructured data. SPARQL provides for users possibility to execute analytics on data stored in a relational DB, as well as friend-of-a-friend (FOAF) relationships, PageRank and shortest path.

Conclusions of the First Part

Modern software engineering industry developed extremely wide range of instruments for implementation of the given task.

Different programming languages may have own advantages and disadvantages, simplicity for learning and range of tools provided. In the same time, almost any task can be implemented using any language. However, depending on language task can be executed on specified hardware.

In addition, there are exist many wide range of libraries, frameworks and platforms that support languages and help to boost development time, providing for developer functions that have already been implemented so that developer has no need to write them again.

For more efficient development process, it is useful to implement existing programming pattern. Firstly, it is necessary to choose the most suitable for given task pattern and follow its principles.

PART 2
WEB-APPLICATION DEVELOPMENT

2.1. Choosing Suitable Pattern, Language and Technologies

Before developing, it is necessary to make decision about technologies, language and programming pattern to choose. Making such decision before developing is important because web-application architecture depends on it.

2.1.1. Programming Pattern Chosen

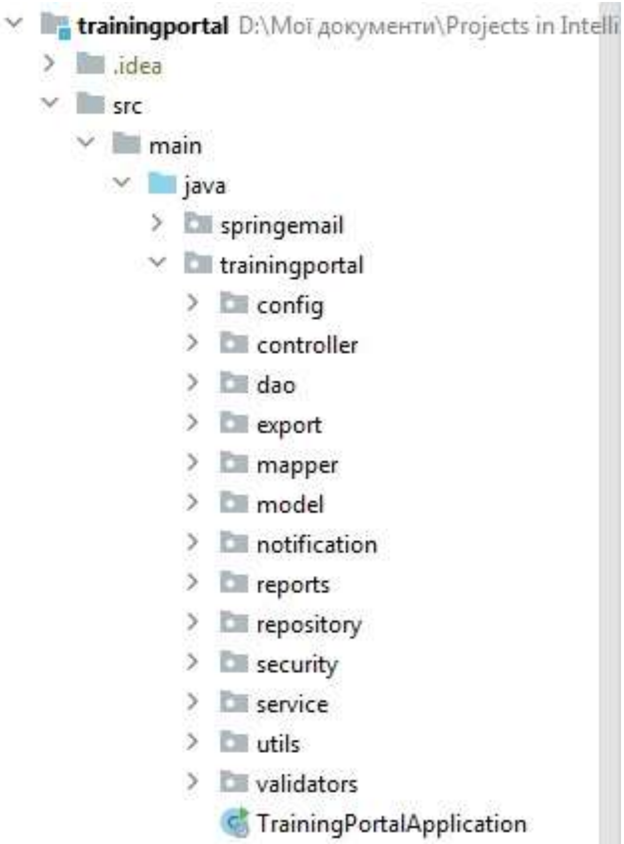
Web-application is a kind of program that runs on server side while the UI runs on client side. In addition, client side part and server side part must effectively communicate with each other.

For effective performing of all parts of web-application, it is necessary to implement suitable programming pattern for such kinds of program. In case of web-application MVC design is suitable. MVC is for Model-View-Controller, the main principle is to divide every part (user appearance, business logic and controller) in separate parts.

Such division is very important for scalability, reliability and simplicity. While every part is separated, it is easier to implement changes into application. For example, business logic code is placed in its own folder (or package) named as service. Due to the its name for developers it is easier to understand the purpose of code situated in this folder. Next, changes provided in the service package (adding one more business logic component or editing existing) are isolated from another packages.

Other parts of application are placed in the same manner. Usually code of controller part placed in folder (package) named as controller. The same is for model.

Controller parts regulates data transfer between client side and server side. It receive http requests from browser and depending on it can call necessary functions in business logic. After receiving answer from called functions, it builds view part and sends it back to the browser (fig. 2.1).



```
trainingportal D:\Мої документи\Projects in Intelli
  .idea
  src
    main
      java
        springemail
        trainingportal
          config
          controller
          dao
          export
          mapper
          model
          notification
          reports
          repository
          security
          service
          utils
          validators
          TrainingPortalApplication
```

Fig. 2.1. Backend project architecture with service and model packages

Model is part of web-application that contains structure of used tables in database. Usually folder has the same name. This code is necessary to represent tables in program. It allows creating connection with database and providing data manipulation with between database and application.

2.1.2. Programming Language Chosen

Generally, it is possible to perform any task with any programming language it is possible to perform any task. However, it differs on what hardware task should be executed.

In the case of web-application two programming languages were chosen. Backend part is written on Java and some frontend part is written on JavaScript.

Java is the most popular language among those that can be executed on server side. There are exist dozens of libraries and frameworks for simplifying of development. Due to popularity, Java has a big community that helps to get information about language according to operational needs. Java language is often used for developing applications of enterprise levels. Most hosting providers also support it. Java language provide concept of classes. This helps to build structural project, simplify searching necessary element among dozens of code rows.

Java is high-level language that in the same time powerful enough. It is platform independent.

JavaScript is the most popular frontend language. It is executed on the client side (on browser). JavaScript is used for creating dynamic pages. It provides possibilities to design dynamic layout of different elements such as appearance of buttons, changing their colors and so on.

2.1.3. Frameworks Chosen

Among most popular frameworks for Java, Spring is placed on the high rank. Spring framework provides a huge variety of possibilities. It is enormous by itself. So that it was created subdivision of Spring framework – Spring Boot framework. Spring Boot is lighter and contains all necessary instruments for fast application setup.

It allows to implement security – authorization mechanism, sessions supporting and user role management. It allows to open and close access to different parts of application in accordance of logined user’s role.

Spring Boot provides powerful instrument of dependency injection using annotations. After annotating necessary Java classes it connect them among each other by itself. In addition, Spring Boot is provided by auto configuration mechanism that allows developer not to spend his time for configuring. This feature differs Spring Boot from Spring that require manual configuration for starting working.

Due to popularity of Spring Boot it has great support of developers community. In addition, developer’s website provide big variety of tutorials for using Spring Boot in different use cases.

In addition, Spring Boot Framework has useful instruments for implementing necessary mechanisms for creating of web application. It is very suitable for creating web apps according Model-View-Controller (MVC) pattern (as part of Spring MVC framework that is part of Spring Framework). Spring Boot provides mechanisms such as dependency injections and inversion of control. Using annotations and searching through standard hierarchy of project structure it framework allows to unite classes of controller, service and so on to make it work as one system. It is possible to create the same things using Java Core. However, Spring Boot much simplify developing allowing to create web application much faster due to reducing amount of code that it would necessary for developer to write.

For pages layout designing CSS is used. However, there are also CSS frameworks exist. One of the best of them is Bootstrap [19]. It provides classes that simplify pages designing. Adding such class it is used predefined CSS for making changes to layout. In addition, Bootstrap contains some JavaScript libraries for adding dynamics to pages by simple class adding.

Bootstrap can be added to html page as simple link to the Bootstrap server. Such variant has pros and cons. Pros is that project is lighter and it is easier to share it. Cons is that executing of project depends on Internet connection and Bootstrap server availability. Another variant is to download whole framework and import it into the project. At the same time such decision also has its own pros and cons that are opposite to the previous variant of simple link adding.

Bootstrap also has its general cons that it is possible to create only simple layouts with its own libraries. In the case when some attractive design is necessary, additional custom CSS is needed. However, it is not a problem to use own custom CSS with Bootstrap simultaneously. In the case when simple website design is acceptable, Bootstrap is great decision for fast up of developing process.

As additional great pros is that Bootstrap was designed with flexibility in mind. It is easy enough to specify different page layout depending on screen size. It allows making pages both suitable for big PC screens and for small screens of mobile devices. Bootstrap provides different levels of screen sizes so that layout can be on small screen in one way, on middle screen in another way and on big screens in one more way.

2.1.4. Database Chosen

Among existing databases, most common for such purposes is relational databases. At the same time, it exists different relational databases, developed by different companies and distributed as free or as proprietary.

Most famous relational databases are Oracle DB, MySQL, Microsoft SQL Server (fig. 2.2) [20]. Oracle DB not only the famous but also provide good structured documentation that makes it simpler to start with it.

Different databases are work with different dialects of SQL. So that documentation support is important for providing better understanding of used SQL dialect features. Oracle DB uses PL/SQL. It understands all SQL features; however, it also has its own.

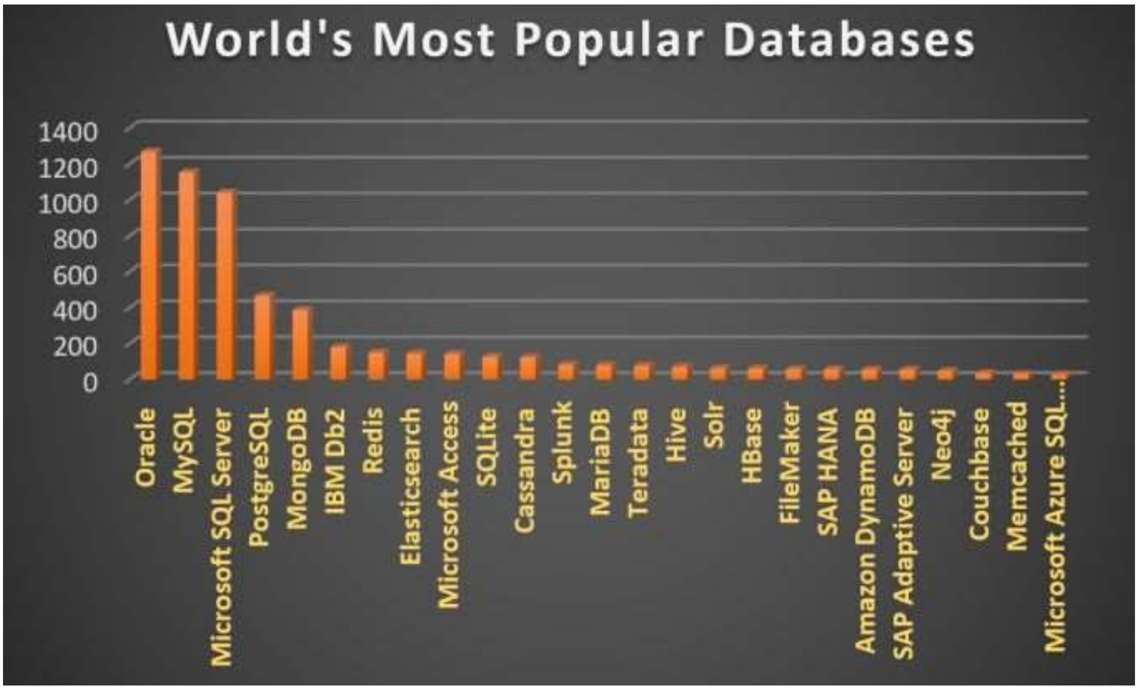


Fig. 2.2. Most popular Databases

At the same time, Oracle DB (express edition) is free for use (for not commercial purposes). RDMS can be easily downloaded from Oracle website. However, it needs registration.

2.1.5. Additional Tools

In any web application, frontend part and backend part should interact with each other. At the same time, it is common situation when frontend part must be changeable depending on server outputs. To have ability show variable content, some template engines exist. Such template engines allow simplifying process. One of them is Thymeleaf.

Thymeleaf is pronounced like this: /`taimli:f/ [21]. This is template engine that is easily to implement into the project. It is provided with good explained documentation and executed on Java. It is executed on server side. However, code is written inside HTML tags, after executing on server side, it sends on frontend only results of execution as static HTML code.

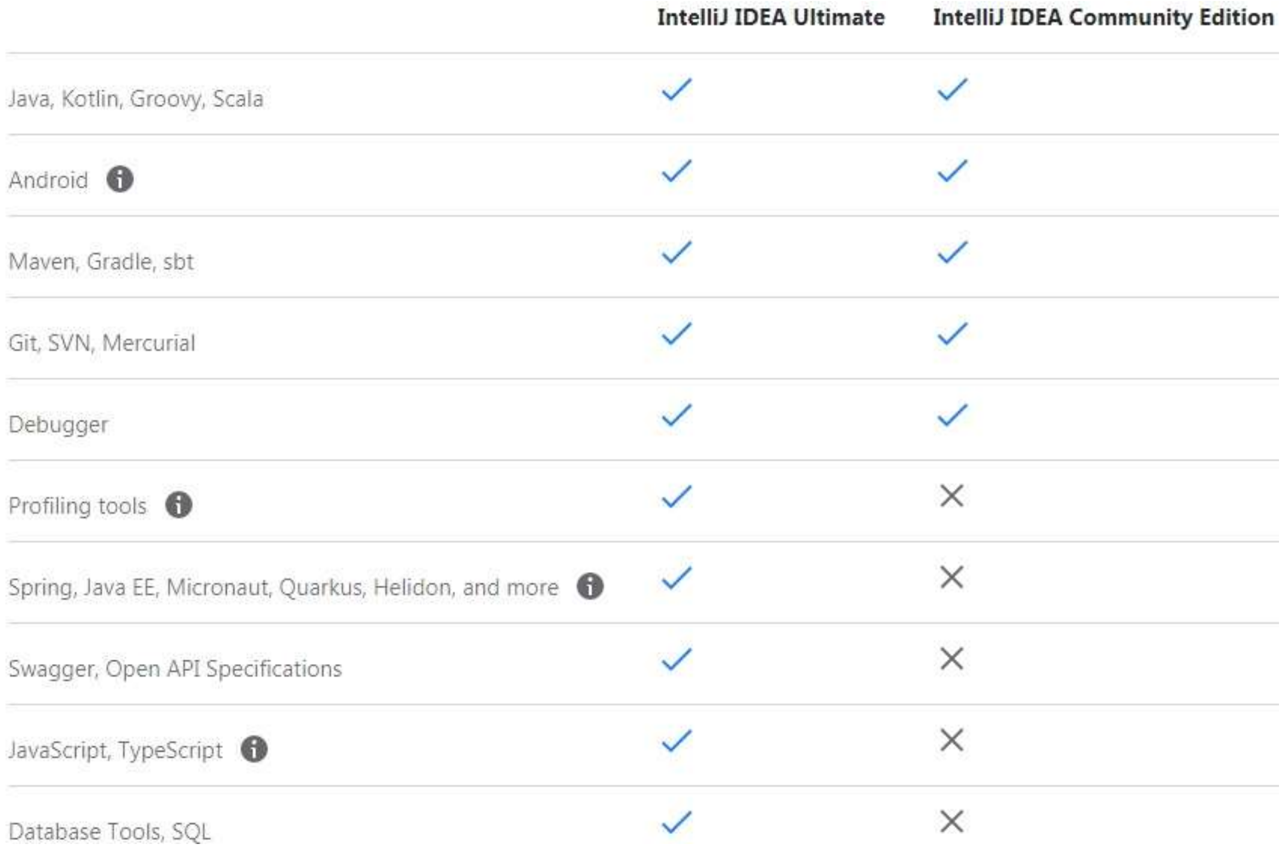
2.2. Preparing Tools for Web Application Designing

Designing of web application is very complex procedure. It requires usage of huge variety of technologies and tools, performing software settings and knowledge of basics with ability to search necessary information in the Internet.

Designing of web application requires not only knowledge of languages, frameworks, patterns (and other technologies for development) but also knowledge of tools like IDEs, RDMS settings, connecting IDEs with DB and ability to solve issues that may occur while preparing to start developing.

2.2.1. Preparing IDE for Developing

Firstly, IDE should be installed and prepared for work. There are many different IDEs. In this case, IntelliJ IDEA was chosen. It provides intellectual features to make developing easier. There are two editions: Ultimate and Community Edition (fig. 2.3) [22].

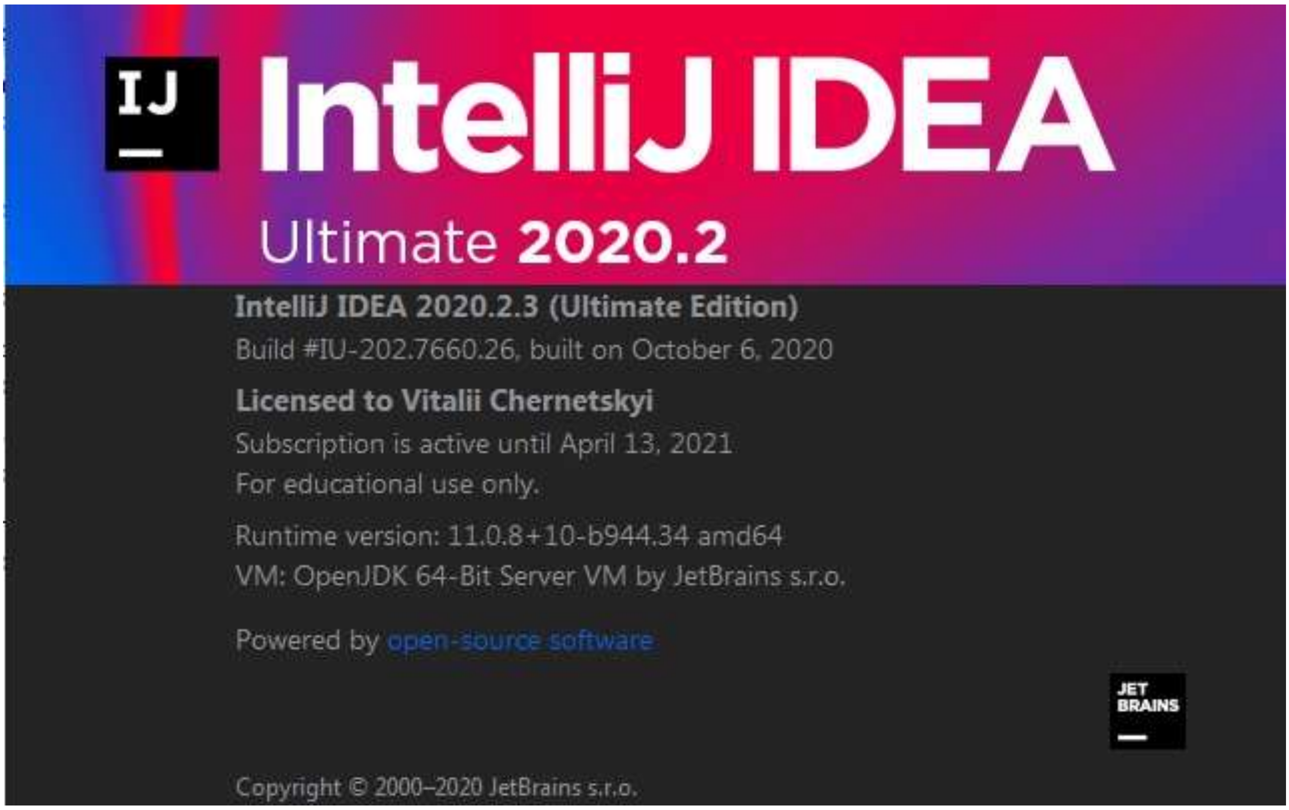


	IntelliJ IDEA Ultimate	IntelliJ IDEA Community Edition
Java, Kotlin, Groovy, Scala	✓	✓
Android	✓	✓
Maven, Gradle, sbt	✓	✓
Git, SVN, Mercurial	✓	✓
Debugger	✓	✓
Profiling tools	✓	✕
Spring, Java EE, Micronaut, Quarkus, Helidon, and more	✓	✕
Swagger, Open API Specifications	✓	✕
JavaScript, TypeScript	✓	✕
Database Tools, SQL	✓	✕

Fig. 2.3. Comparing Ultimate and Community IDEA editions

Because of use of Spring Boot framework and JavaScript only Ultimate edition passes for the project. However, IntelliJ IDEA provides free student license for Ultimate edition (fig. 2.4).

First set up requires simple settings that can be executed by following default flow of first set up.



IntelliJ IDEA 2020.2.3 (Ultimate Edition)
Build #IU-202.7660.26, built on October 6, 2020

Licensed to Vitalii Chernetskyi
Subscription is active until April 13, 2021
For educational use only.

Runtime version: 11.0.8+10-b944.34 amd64
VM: OpenJDK 64-Bit Server VM by JetBrains s.r.o.

Powered by open-source software

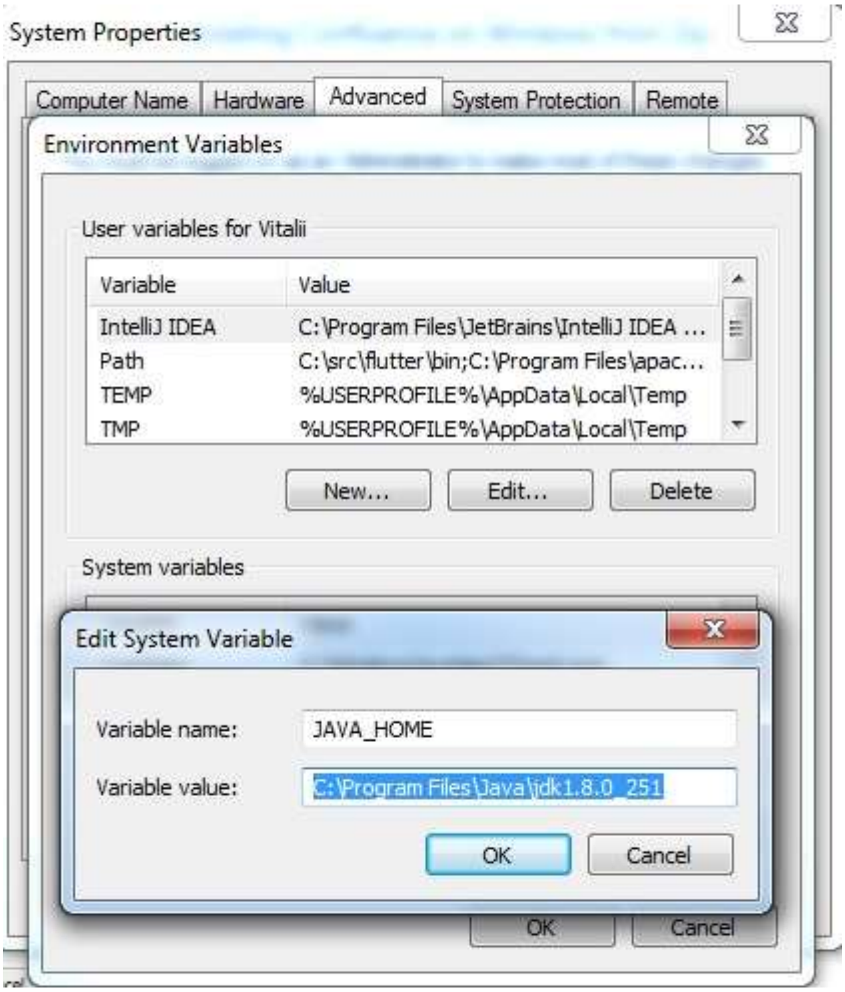
Copyright © 2000–2020 JetBrains s.r.o.

Fig. 2.4. Information about IntelliJ IDEA installed for web app developing

Creating new project requires initial settings. Project correct set up helps to avoid some issues. There are several method of creating new projects. Simple Java project or using Spring Initializr. Spring Initializr is provided by Spring Framework to create project with necessary dependencies for quick startup of web application. It allows to choose what Spring Boot libraries is necessary to connect with project, and some libraries from another providers.

Firstly, it is necessary to choose Java version for the project. In this case, Java 8 was chosen (officially, eighth version is named 1.8). To have ability develop project with Java 8 it is necessary to install Java Development Kit (JDK) of necessary version. JDK includes a complete Java Runtime Environment (JRE) plus tools for developing, debugging, and monitoring Java applications.

After installing JDK, some additional settings on OS should be performed. As this project was developed on Windows 7 OS, given settings are related particularly to this version of OS. System variable JAVA_HOME must be specified with path to JDK. It is possible to make by following next path: My Computer and select Properties > Advanced. Then click the Environment Variables button. Under System Variables, click New. In the Variable Name field, enter JAVA_HOME and in Variable Field enter JDK installation path (fig. 2.5).



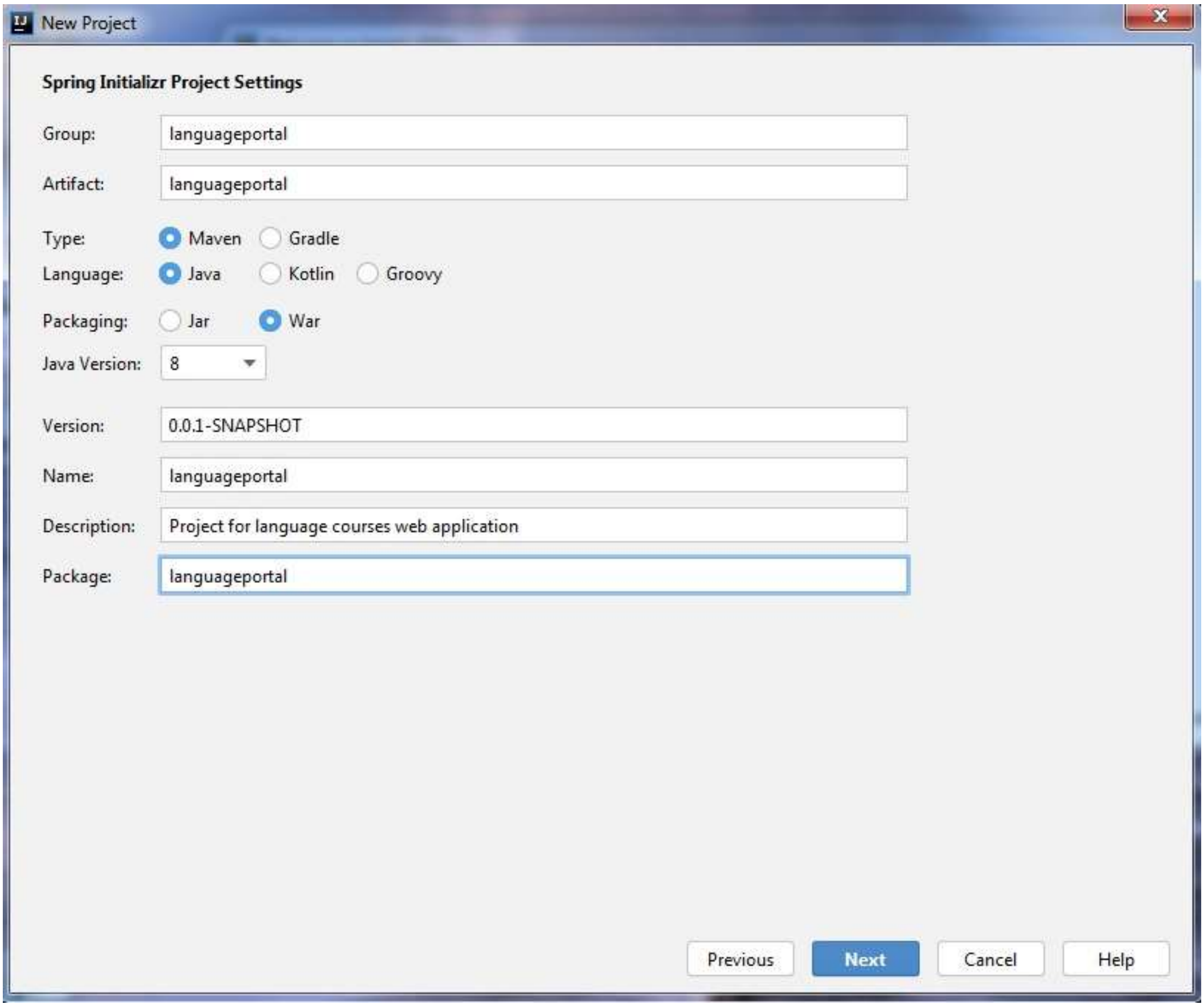
Variable	Value
IntelliJ IDEA	C:\Program Files\JetBrains\IntelliJ IDEA ...
Path	C:\src\flutter\bin;C:\Program Files\apac...
TEMP	%USERPROFILE%\AppData\Local\Temp
TMP	%USERPROFILE%\AppData\Local\Temp

Variable name: JAVA_HOME

Variable value: C:\Program Files\Java\jdk1.8.0_251

Fig. 2.5. Edit System Variable window for preparing JDK usage

After creating JAVA_HOME variable and assigning JDK installation path, it becomes possible to create project in IntelliJ IDEA using installed JDK. Creating project with Spring Initializr also needs to have active Internet connection because it issues Spring server for generation of new project. Then initial settings of new project should be performed (fig. 2.6).



New Project

Spring Initializr Project Settings

Group: languageportal

Artifact: languageportal

Type: ● Maven ○ Gradle

Language: ● Java ○ Kotlin ○ Groovy

Packaging: ○ Jar ● War

Java Version: 8

Version: 0.0.1-SNAPSHOT

Name: languageportal

Description: Project for language courses web application

Package: languageportal

Fig. 2.6. Spring Initializr project settings

Field “Group”: project coordinates. Also infers the root package name to use.

Field “Artifact”: project coordinates. Also infers the name of the project.

Field “Name”: display name of the project that also determines the name of Spring Boot application. For instance, if the name of project is my-app, the generated project will have a MyAppApplication class.

Field “Description”: description of the project.

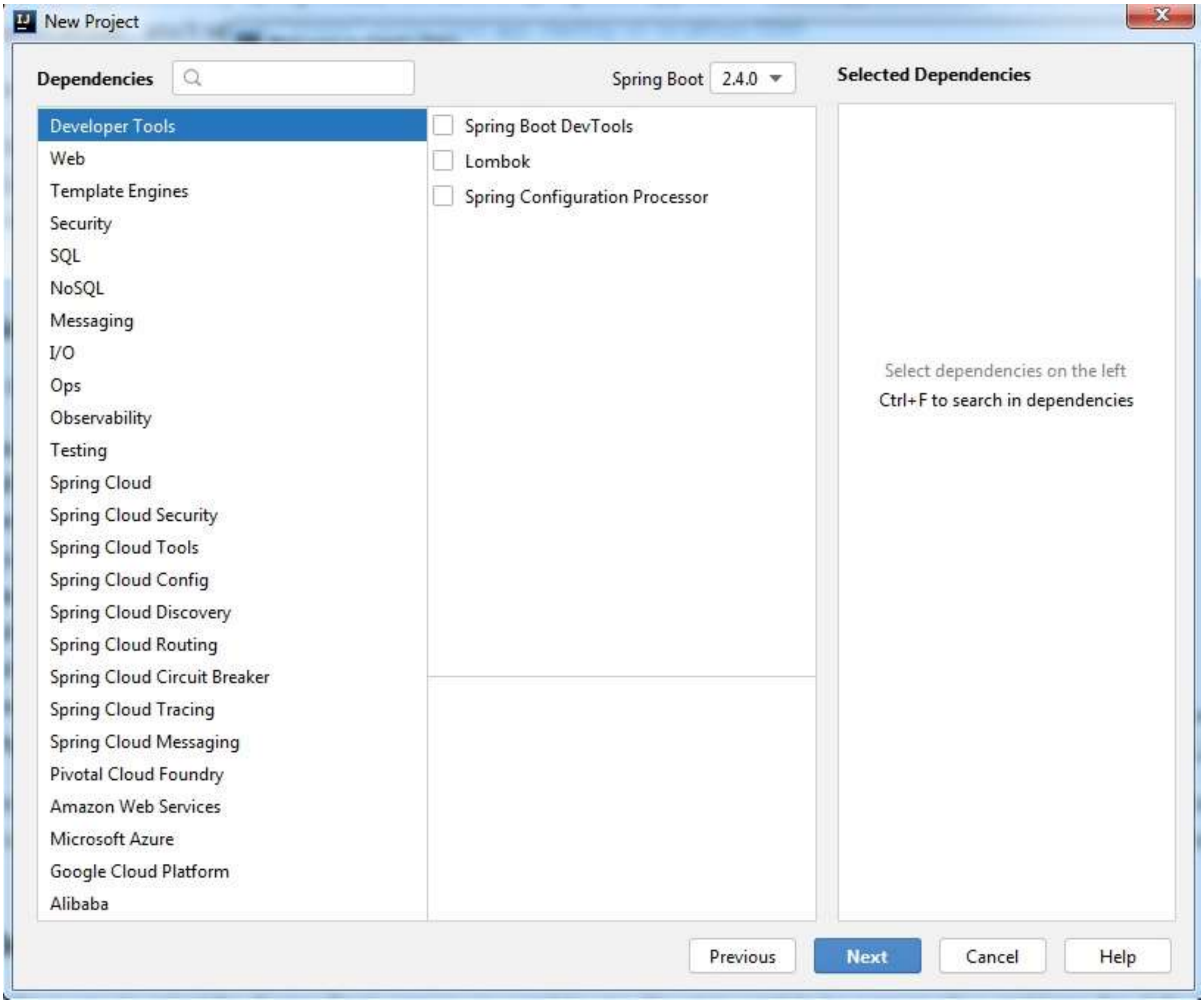
Field “Package”: root package of the project. If not specified, the value of the Group attribute is used.

Field “Packaging”: project packaging. It can be generated as jar or war projects.

Field “Java Version”: the Java version to use.

Field “Language”: the programming language to use.

After making decisions in initial settings window, it is necessary to make decisions about dependencies to connect (fig. 2.7).



Spring Boot 2.4.0

□ Spring Boot DevTools
□ Lombok
□ Spring Configuration Processor

Select dependencies on the left
Ctrl+F to search in dependencies

Fig. 2.7. Available dependencies to connect in Spring Boot project

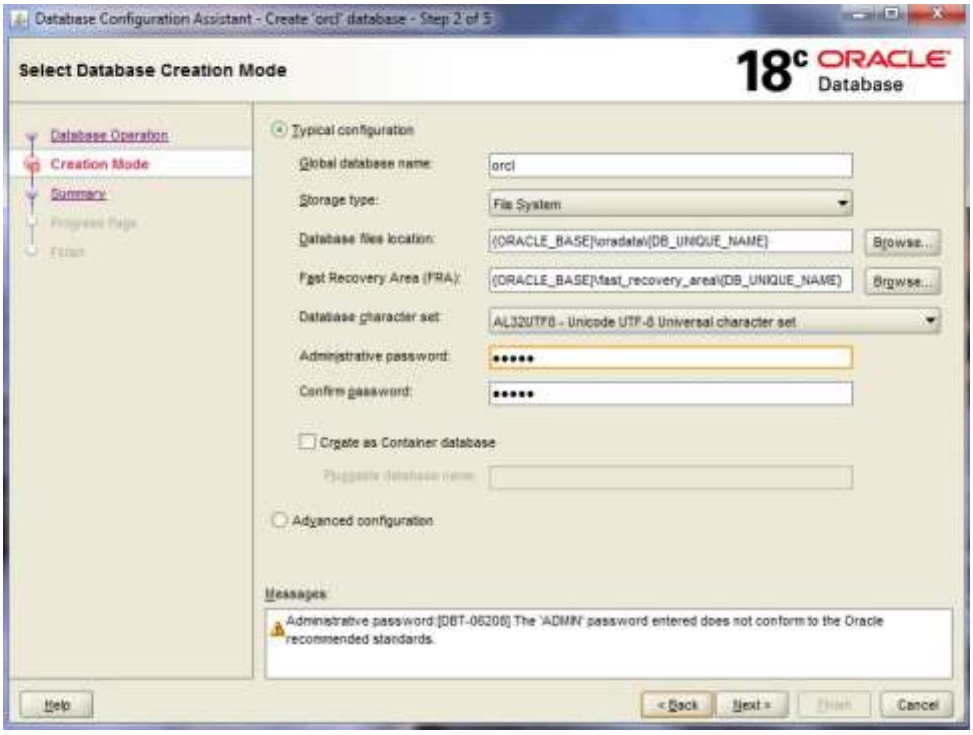
Among available dependencies for this project, such decisions were made. Spring Boot DevTools. It provides fast application restarts, LiveReload and configurations for enhanced development experience. Lombok. It is Java annotation library which helps to reduce boilerplate code. Spring Web. It is necessary for building web, including RESTful, applications using Spring MVC. Uses Apache Tomcat as the default embedded container. Spring Session. It provides an API and implementations for managing user session information. Thymeleaf. It is a modern server-side Java template engine for both web and standalone environments. Allows HTML to be correctly displayed in browsers and as static prototypes. Spring Security. It is highly customizable authentication and access-control framework for Spring applications. JDBC API. It is Database Connectivity API that defines how a client may connect and query a database. Spring Data JPA. Persist data in SQL stores with Java Persistence API using Spring Data and Hibernate. Oracle Driver. It is a JDBC driver that provides access to Oracle. Java Mail Sender. For sending email using Java Mail and Spring Framework's JavaMailSender.

Next, it is necessary to choose location for storing project and then finish creation.

2.2.2. Preparing Database for Developing

In this project, Oracle DB is used. Oracle company provides some free Express version of its own database that is possible to download from their website. For performing downloading, it is necessary register on Oracle website.

After downloading, it is necessary to perform procedure of installation. It requires some settings to perform for correct installation and database instance creation (fig. 2.8).



Database Configuration Assistant - Create 'orcl' database - Step 2 of 5

Select Database Creation Mode

● Typical configuration

Global database name: orcl

Storage type: File System

Database files location: {ORACLE_BASE}\oradata\{DB_UNIQUE_NAME}

Fast Recovery Area (FRA): {ORACLE_BASE}\fast_recovery_area\{DB_UNIQUE_NAME}

Database character set: AL32UTF8 - Unicode UTF-8 Universal character set

Administrative password: •••••

Confirm password: •••••

□ Create as Container database

Pluggable database name:

○ Advanced configuration

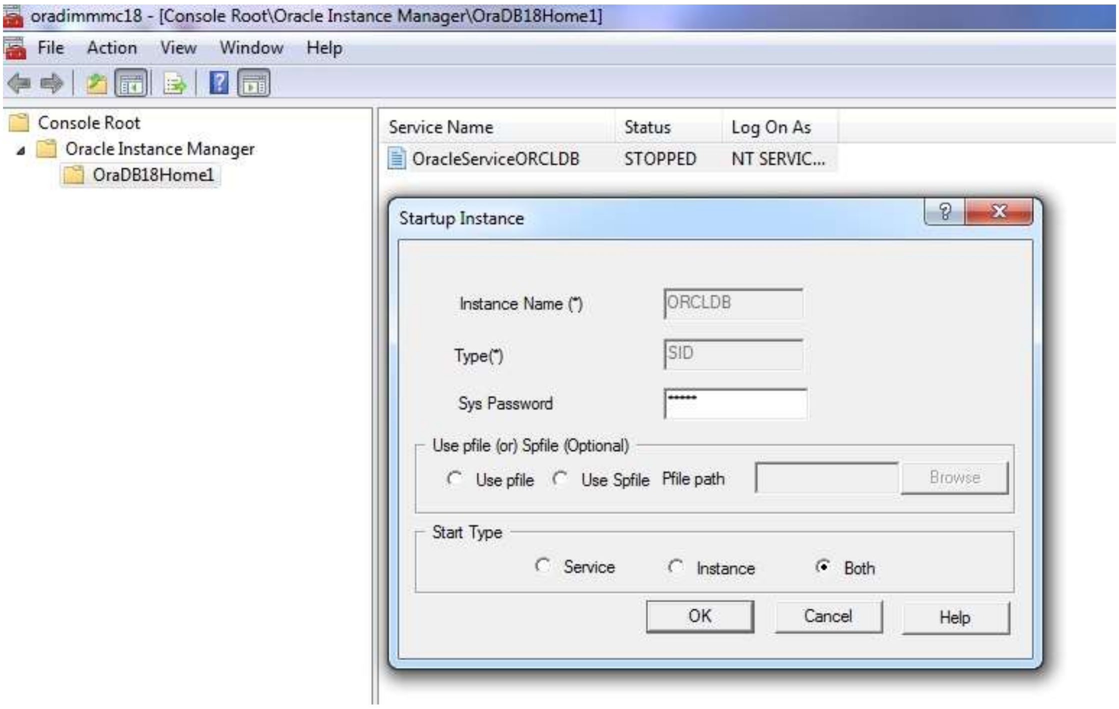
Messages:

Administrative password:[DBT-06208] The 'ADMIN' password entered does not conform to the Oracle recommended standards.

Fig. 2.8. Window of database creation mode

While creation it is important to create own authentication password (instead of using OS authentication). In case of using OS authentication there will be problems in connecting web application to database. Password should be strong enough. In case of weak password, there will be warning with advice to change password. However, it is not forbidden to ignore warning and continue.

After database instance was created, it can be with manual setup (instead of auto startup). For controlling of instance running special Oracle app is used – Oracle Instance Manager. It provides possibilities to control different regimes of database instance startup (fig. 2.9).



Service Name	Status	Log On As
OracleServiceORCLDB	STOPPED	NT SERVIC...

Instance Name (*) ORCLDB

Type(*) SID

Sys Password *****

Use pfile (or) Spfile (Optional)

○ Use pfile ○ Use Spfile Pfile path

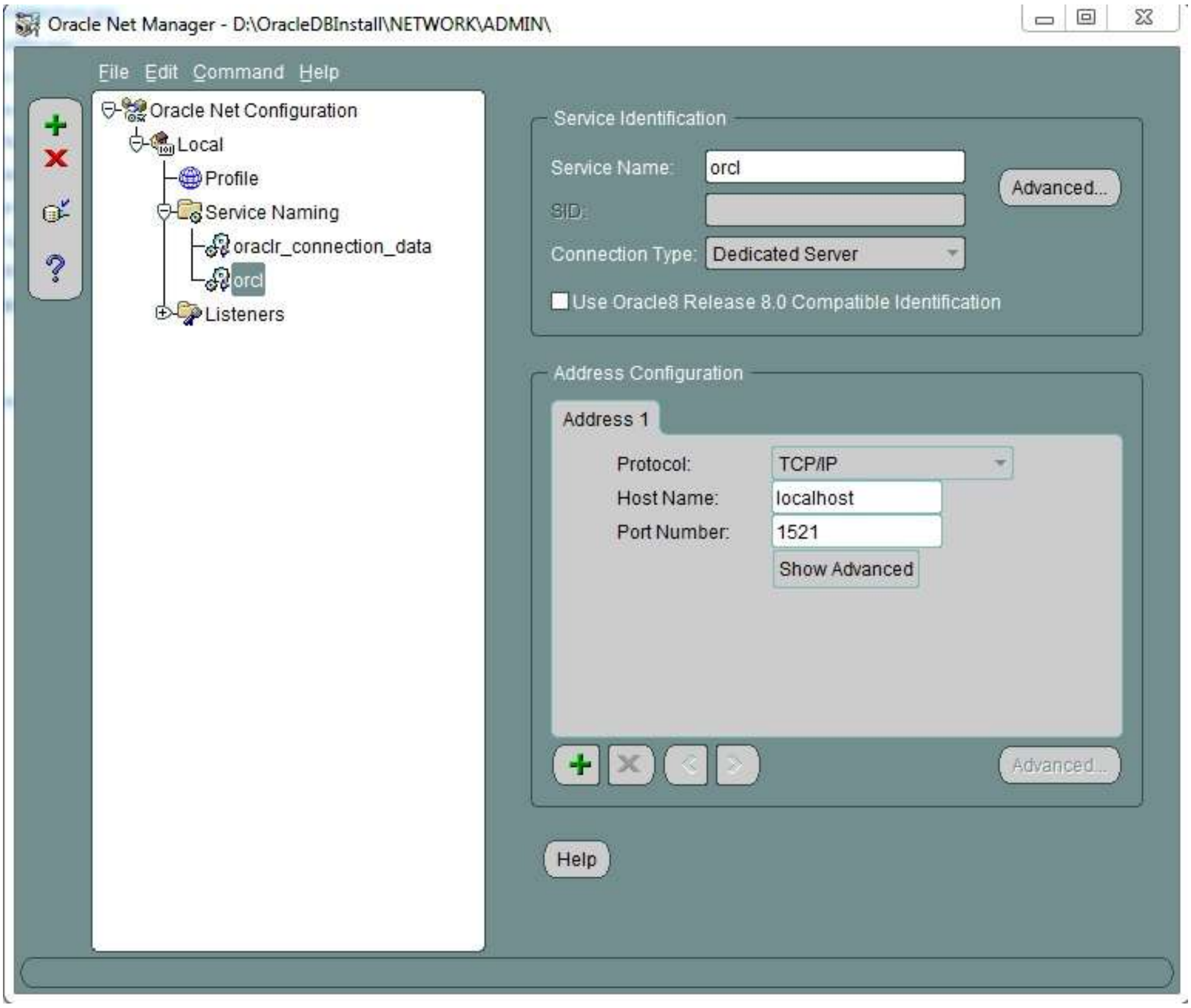
Start Type

○ Service ○ Instance ● Both

Fig. 2.9. Window Startup Instance in Oracle Instance Manager

Field Sys Password should be filled with password defined while database instance was creating. It takes some time to startup. However, during startup procedure can be illusion of program freezing.

For connecting of web application with database instance, network port is used. By default, Oracle database works on 1521 port. However, it can be checked and changed with the help of special Oracle program – Net Manager. Following the path Oracle Net Configuration -> Local -> Service Naming -> [created instance name] (fig. 2.10).



Service Name: orcl

SID:

Connection Type: Dedicated Server

☐ Use Oracle8 Release 8.0 Compatible Identification

Protocol: TCP/IP

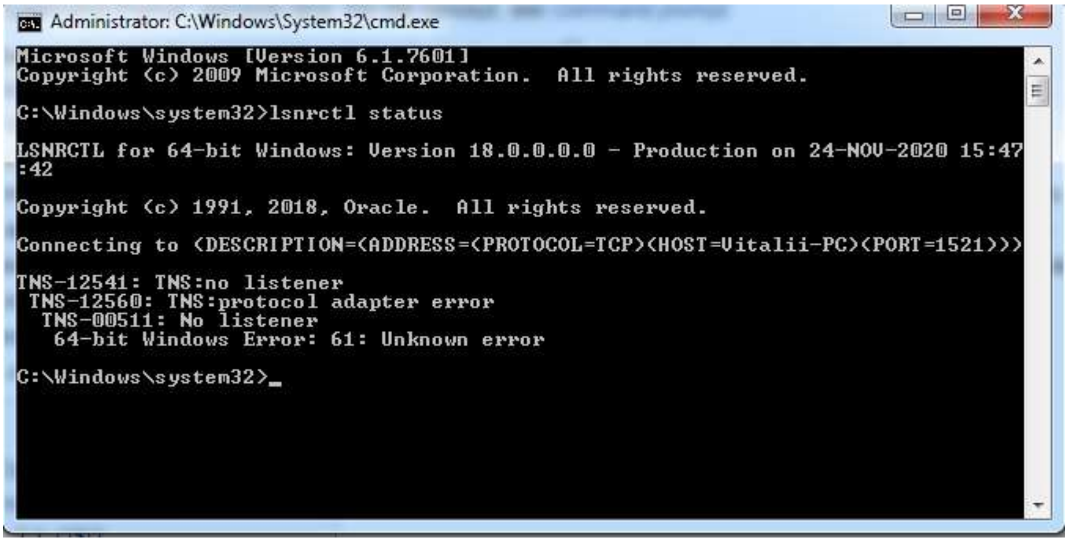
Host Name: localhost

Port Number: 1521

Fig. 2.10. Window of Net Manager with instance Address Configuration

In addition, to have possibility of connection to Oracle database special Listener Control utility should be started. Listener Control is an SQL*Net utility used for controlling database listeners. A listener is required for allowing remote (not local) clients to connect to the Oracle database via the network. This utility cannot create or configure listeners, but provides commands to control listener functions such as starting and stopping listeners, reporting the status of listeners, changing parameter listener settings, and so on [23]. The Oracle Listener is a process listening for incoming database connections. This process is only needed on the database server side. The listener is controlled via the lsnrctl utility. Configuration is done via the listener.ora file [24]. Usually it starts automatically. However, sometimes it is necessary to start it manually.

To check if Listener Control started special command should be executed in cmd (command-line interpreter) in Windows. There are important moment – cmd must be opened with administrative rights. Without administrative rights necessary commands will output errors. For checking if Listener Control started lsnrctl status command should be executed (fig. 2.11).

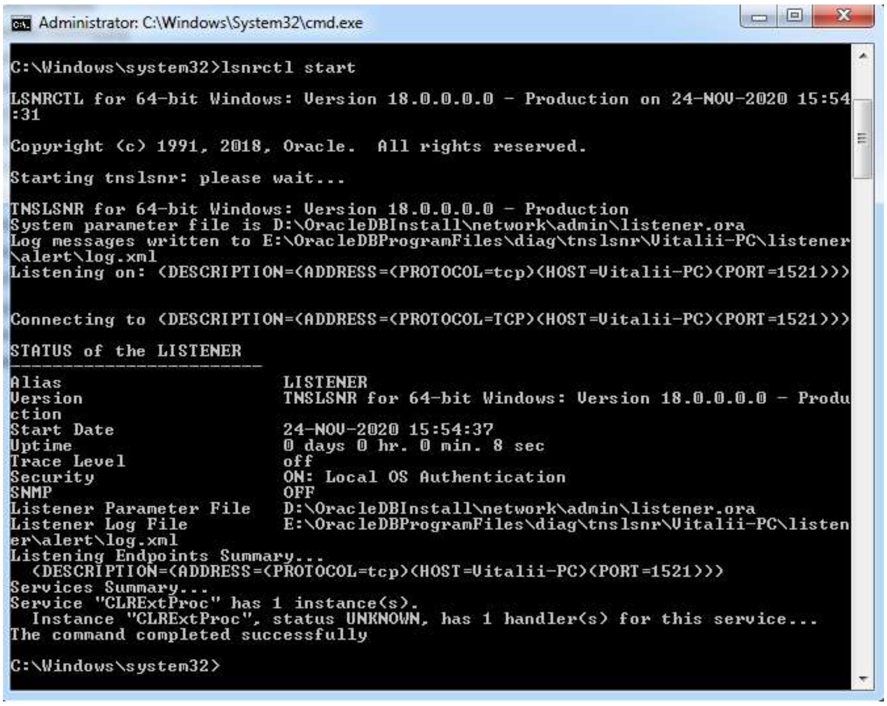


Administrator: C:\Windows\System32\cmd.exe

```
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation.  All rights reserved.

C:\Windows\system32>lsnrctl status

LSNRCTL for 64-bit Windows: Version 18.0.0.0.0 - Production on 24-NOV-2020 15:47
:42

Copyright (c) 1991, 2018, Oracle.  All rights reserved.

Connecting to (DESCRIPTION=(ADDRESS=(PROTOCOL=TCP)(HOST=Vitalii-PC)(PORT=1521)))

TNS-12541: TNS:no listener
 TNS-12560: TNS:protocol adapter error
  TNS-00511: No listener
   64-bit Windows Error: 61: Unknown error

C:\Windows\system32>_
```

Fig. 2.11. Result of lsnrctl status command executed with Listener Control not started

Execution results show that there are no listener, protocol adapter error and other errors. All these errors are because of Listener Control utility is not working. To start Listener Control lsnrctl start command must be executed (fig. 2.12).

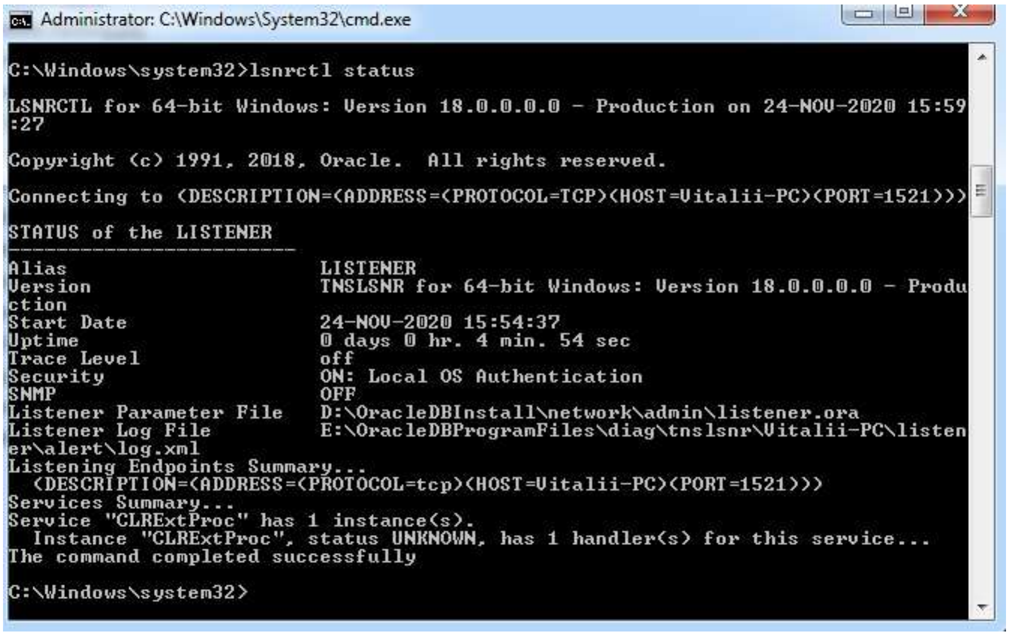


Administrator: C:\Windows\System32\cmd.exe

```
C:\Windows\system32>lsnrctl start

LSNRCTL for 64-bit Windows: Version 18.0.0.0.0 - Production on 24-NOV-2020 15:54
:31

Copyright (c) 1991, 2018, Oracle.  All rights reserved.

Starting tnslsnr: please wait...

TNSLSNR for 64-bit Windows: Version 18.0.0.0.0 - Production
System parameter file is D:\OracleDBInstall\network\admin\listener.ora
Log messages written to E:\OracleDBProgramFiles\diag\tnslsnr\Vitalii-PC\listener
\alert\log.xml
Listening on: (DESCRIPTION=(ADDRESS=(PROTOCOL=tcp)(HOST=Vitalii-PC)(PORT=1521)))


Connecting to (DESCRIPTION=(ADDRESS=(PROTOCOL=TCP)(HOST=Vitalii-PC)(PORT=1521)))
STATUS of the LISTENER
------------------------
Alias                     LISTENER
Version                   TNSLSNR for 64-bit Windows: Version 18.0.0.0.0 - Produ
ction
Start Date                24-NOV-2020 15:54:37
Uptime                    0 days 0 hr. 0 min. 8 sec
Trace Level               off
Security                  ON: Local OS Authentication
SNMP                      OFF
Listener Parameter File   D:\OracleDBInstall\network\admin\listener.ora
Listener Log File         E:\OracleDBProgramFiles\diag\tnslsnr\Vitalii-PC\listen
er\alert\log.xml
Listening Endpoints Summary...
  (DESCRIPTION=(ADDRESS=(PROTOCOL=tcp)(HOST=Vitalii-PC)(PORT=1521)))
Services Summary...
Service "CLRExtProc" has 1 instance(s).
  Instance "CLRExtProc", status UNKNOWN, has 1 handler(s) for this service...
The command completed successfully

C:\Windows\system32>
```

Fig. 2.12. Result of lsncrtl start command execution

In last line of output result is printed “The command completed successfully” which means that Listener Control utility started without problems. It is possible to check again Listener Control availability with lsncrtl status command (fig. 2.13).



```
C:\Windows\system32>lsnrctl status

LSNRCTL for 64-bit Windows: Version 18.0.0.0.0 - Production on 24-NOV-2020 15:59
:27

Copyright (c) 1991, 2018, Oracle.  All rights reserved.

Connecting to (DESCRIPTION=(ADDRESS=(PROTOCOL=TCP)(HOST=Vitalii-PC)(PORT=1521)))
STATUS of the LISTENER
------------------------
Alias                     LISTENER
Version                   TNSLSNR for 64-bit Windows: Version 18.0.0.0.0 - Produ
ction
Start Date                24-NOV-2020 15:54:37
Uptime                    0 days 0 hr. 4 min. 54 sec
Trace Level               off
Security                  ON: Local OS Authentication
SNMP                      OFF
Listener Parameter File   D:\OracleDBInstall\network\admin\listener.ora
Listener Log File         E:\OracleDBProgramFiles\diag\tnslsnr\Vitalii-PC\listen
er\alert\log.xml
Listening Endpoints Summary...
  (DESCRIPTION=(ADDRESS=(PROTOCOL=tcp)(HOST=Vitalii-PC)(PORT=1521)))
Services Summary...
Service "CLRExtProc" has 1 instance(s).
  Instance "CLRExtProc", status UNKNOWN, has 1 handler(s) for this service...
The command completed successfully

C:\Windows\system32>
```

Fig. 2.13. Result of lsnrctl status command execution with Listener Control running

When Listener Control is running, there are no errors after executing lsnrctl control command. Instead, there are message “The command completed successfully” which means that Listener Control was checked successfully.

2.3. Web Application Designing

Web application designing requires complex knowledge of different designing aspects. It is necessary to implement different technologies and make them work together. It is necessary to understand project structure to make it easier for solve issues that may appear with application growth.

Every component may have its own position in the project. Programming code for server side, code for frontend side, application properties – all these things have defined position. In the same time, they should “know” each other’s position to interact.

2.3.1. Maven and Project Structure

Maven is a build automation tool used primarily for simplifying project developing. During designing of application, many additional libraries may be required. It is be possible to connect them by manual downloading and including them. However, such solution is problematic because of many actions necessary to perform. In addition, such project becomes harder to share with others because of necessity to copy and send used libraries.

To solve such problem some build automation tools were created. One of them is Maven. Maven database contains a huge amount of references to a big variety of libraries. In the project, configurations are made in specialized file named pom.xml. POM is refers to project object model.

Pom.xml is filled with special tags and data inside them. These data specify name of the project, main properties of it and references to libraries. These references are written inside special <dependency> tag. It should be written in special format. At first name of group ID is written (in special tag) and then artifact ID (also in special tag).

Tag “groupId” uniquely identifies project across all projects. A group ID should follow Java's package name rules. This means it starts with a reversed domain name that developer controls. Maven does not enforce this rule. There are many legacy projects that do not follow this convention and instead use single word group IDs. However, it will be difficult to get a new single word group ID approved for inclusion in the Maven Central repository. It is possible to create as many subgroups as necessary. A good way to determine the granularity of the groupId is to use the project structure. That is, if the current project is a multiple module project, it should append a new identifier to the parent's groupId. Tag “artifactId” is the name of the jar without version. If developer created it, then he can choose whatever name is suitable with lowercase letters and no strange symbols. If it's a third party jar, developer has to take the name of the jar as it is distributed. Tag “version” if developer distribute it, then he can choose any typical version with numbers and dots (1.0, 1.1, 1.0.1, ...). It is not recommended to use dates as they are usually associated with SNAPSHOT (nightly) builds. If it is a third party artifact, developer has to use their version number whatever it is, and as strange as it can look [25].

Java package rules have some principles of naming. Developers should take steps to avoid the possibility of two published packages having the same name by choosing unique package names for packages that are widely distributed. This allows packages to be easily and automatically installed and catalogued. This section specifies a suggested convention for generating such unique package names. Implementations of the Java platform are encouraged to provide automatic support for converting a set of packages from local and casual package names to the unique name format described here.

If unique package names are not used, then package name conflicts may arise far from the point of creation of either of the conflicting packages. This may create a situation that is difficult or impossible for the user or programmer to resolve. The class ClassLoader can be used to isolate packages with the same name from each other in those cases where the packages will have constrained interactions, but not in a way that is transparent to a native program.

It is necessary to form a unique package name by first having (or belonging to an organization that has) an Internet domain name, such as sun.com. Developer then reverse this name, component by component, to obtain, in this example, com.sun, and use this as a prefix for a package names, using a convention developed within developer’s organization to further administer package names.

In some cases, the internet domain name may not be a valid package name. Here are some suggested conventions for dealing with these situations:

1) If the domain name contains a hyphen, or any other special character not allowed in an identifier, it is necessary to convert it into an underscore.

2) If any of the resulting package name components are keywords then it is necessary append underscore to them.

3) If any of the resulting package name components start with a digit, or any other character that is not allowed as an initial character of an identifier, have an underscore prefixed to the component.

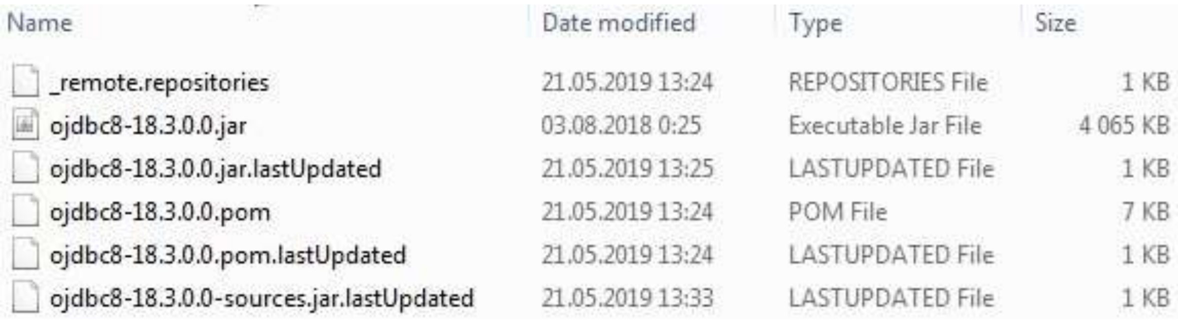
The first component of a unique package name is always written in all-lowercase ASCII letters and should be one of the top level domain names, currently com, edu, gov, mil, net, org, or one of the English two-letter codes identifying countries as specified in ISO Standard 3166, 1981.

The name of a package is not meant to imply where the package is stored within the Internet; for example, a package named edu.cmu.cs.bovik.cheese is not necessarily obtainable from Internet address cmu.edu or from cs.cmu.edu or from bovik.cs.cmu.edu. The suggested convention for generating unique package names is merely a way to piggyback a package naming convention on top of an existing, widely known unique name registry instead of having to create a separate registry for package names [27].

For web application developing, some range of external libraries were used. All of them were connected with the help of Maven system.

Thymeleaf is necessary for creating possibility of dynamic page layout. It executes on server side and sends to browser result as simple HTML code. To connect this library group ID org.springframework.boot with artifact ID spring-boot-starter-thymeleaf were used. In Maven repository documentation, this library specified as starter for building MVC web applications using Thymeleaf views. Next, with the same group ID was used library situated in Maven repository with spring-boot-starter-cache artifact ID. This library provides starter tools for using Spring Framework's caching support. To provide web development possibilities, library with spring-boot-starter-web artifact ID was used. It contains tools for building web, including RESTful, applications using Spring MVC. It uses Tomcat as the default embedded container. Bootstrap framework was also added using Maven repository. Bootstrap is necessary for simplifying creation of user experience layout on page. It uses CSS and JavaScript inside. However, it is easier to use simple class names to specify needed action. With such construction, Bootstrap provides easier solutions. Although, it has poor variety of designing possibilities. To add Bootstrap framework org.webjars group ID was used. Artifact ID has name bootstrap. Maven repository describes that this reference stands for WebJar for Bootstrap. To perform possibility of mail sending, special library was used. This library has group ID com.sun.mail and artifact ID javax.mail. Maven repository names it as JavaMail API. In addition, Spring Boot mail system is used. It is connected with org.springframework.boot group ID and spring-boot-starter-mail artifact ID. Maven repository has such description provided to package: starter for using Java Mail and Spring Framework's email sending support. To provide security, it is easier to use Spring Boot framework. It provides possibility to handle user registration, loginning, logouting, provide roles according to necessity. To connect this library to the project, it is necessary to specify in pom.xml file next dependencies. Group ID should be org.springframework.boot and artifact ID spring-boot-starter-security. Description about package from Maven repository: starter for using Spring Security. There are also obvious package that is necessary to connect. Group ID org.springframework and artifact ID spring-context-support. This package provides Spring Context support. Spring Context provides Application Context, that is Spring's Dependency Injection Container and it is always defined in POMs of artifacts that use Spring Framework. In fact, spring-context depends on spring-core so by defining spring-context as dependency, it is necessary to have spring-core in classpath as well.

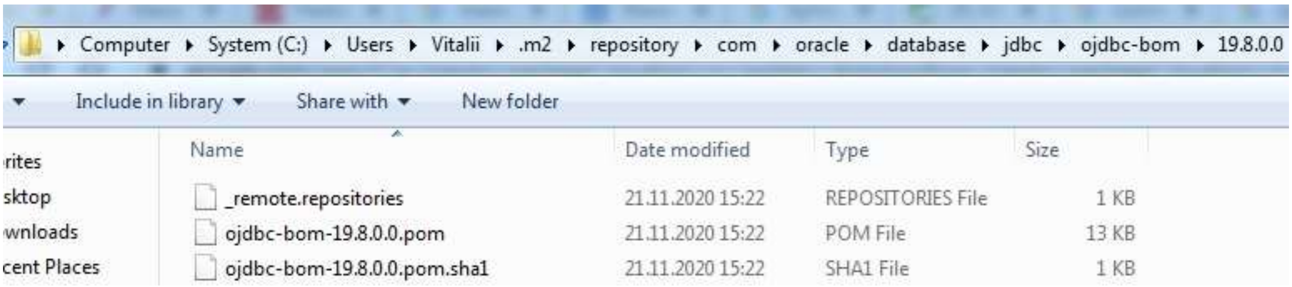
Very important part of necessary external resources is special driver for connecting web application with Oracle database. Theoretically, it can be connected with some Maven dependences. However, in practice it is not easy. For connecting web application with database, Java Database Connectivity (JDBC) driver is used (exactly OJDBC, which stands for Oracle Java Database Connectivity). Problem occurs due to Oracle restriction that prohibits direct download of the driver via Maven repository. It is much easier to download this driver manually and place it in correct position than trying to download it via Maven system. Oracle website provides open free access for manual downloading of ojdbc driver. Oracle provides in one archive some amount of files (fig. 2.14).



Name	Date modified	Type	Size
_remote.repositories	21.05.2019 13:24	REPOSITORIES File	1 KB
ojdbc8-18.3.0.0.jar	03.08.2018 0:25	Executable Jar File	4 065 KB
ojdbc8-18.3.0.0.jar.lastUpdated	21.05.2019 13:25	LASTUPDATED File	1 KB
ojdbc8-18.3.0.0.pom	21.05.2019 13:24	POM File	7 KB
ojdbc8-18.3.0.0.pom.lastUpdated	21.05.2019 13:24	LASTUPDATED File	1 KB
ojdbc8-18.3.0.0-sources.jar.lastUpdated	21.05.2019 13:33	LASTUPDATED File	1 KB

Fig. 2.14. The screenshot of files contained within archive

However, only one file is necessary for the project. It names ojdbc8-18.3.0.0.jar. Numbers in name refer to version of Oracle database that is supported by driver. Next, it is necessary to copy chosen file to the special Maven directory on the host (fig. 2.15).



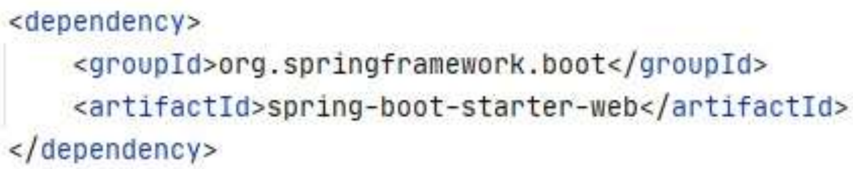
Computer ▸ System (C:) ▸ Users ▸ Vitalii ▸ .m2 ▸ repository ▸ com ▸ oracle ▸ database ▸ jdbc ▸ ojdbc-bom ▸ 19.8.0.0
Include in library
Share with
New folder
rites
sktop
wnloads
cent Places

Name	Date modified	Type	Size
_remote.repositories	21.11.2020 15:22	REPOSITORIES File	1 KB
ojdbc-bom-19.8.0.0.pom	21.11.2020 15:22	POM File	13 KB
ojdbc-bom-19.8.0.0.pom.sha1	21.11.2020 15:22	SHA1 File	1 KB

Fig. 2.15. Location of OJDBC driver

Maven repository usually located in System(C:) -> Users -> [User Name] -> .m2 -> repository -> com -> oracle -> database -> jdbc -> ojdbc-bom -> [version number]. At the same time, in pom.xml file is necessary to make reference to the cloud repository with this driver. Group ID is com.oracle, artifact ID is ojdbc. OJDBC driver will not be downloaded from given resource. However, Maven will use file that was previously copied manually.

Example of creating reference to dependency is shown below (fig. 2.16).

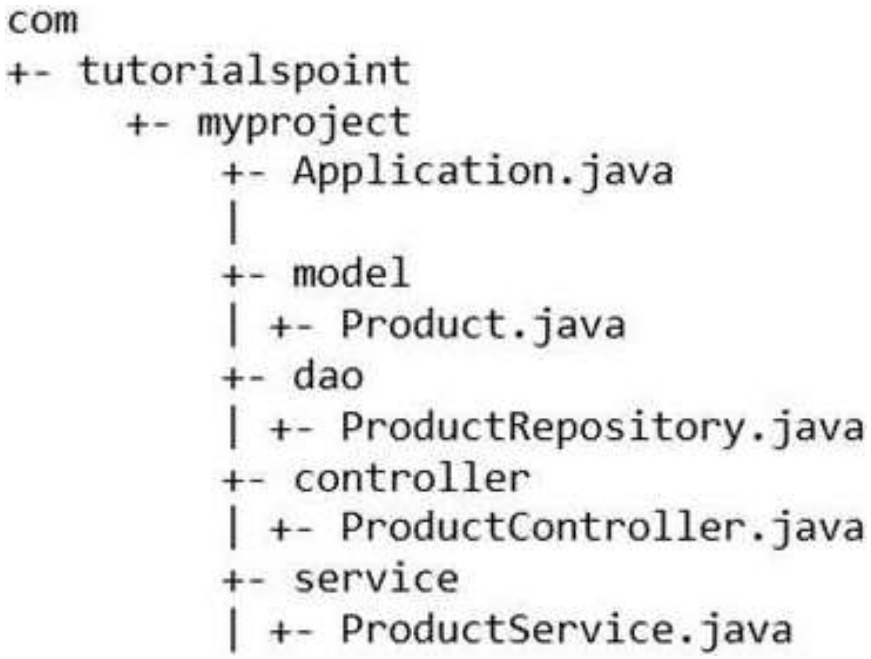


```
<dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-web</artifactId>
</dependency>
```

Fig. 2.16. Example of dependency tag usage

Structure of the whole project is important aspect. It is necessary not only for simpler understanding for developer. Also for correct working of Spring Boot framework. Spring Boot framework has automatic configuration for project classes interconnecting. However, it follows predefined rules of classes and package location in the project.

According to Spring Boot recommendations, typical project structure layout is shown below (fig. 2.17).



```
com
+- tutorialspoint
     +- myproject
         +- Application.java
         |
         +- model
         | +- Product.java
         +- dao
         | +- ProductRepository.java
         +- controller
         | +- ProductController.java
         +- service
         | +- ProductService.java
```

Fig. 2.17. Scheme of the recommended project structure

In addition, the ApplicationName.java file should declare the main method along with @SpringBootApplication annotation.

2.3.2. Implementing MVC Design Pattern

MVC design pattern stands for Model-View-Controller. According to its principles, business logic must be separated from other parts. The same for controlling incoming requests methods with methods to design output layout.

For simplicity of understanding, packages should have corresponding names. So that, for controller methods controller package was created. Every controller class must be annotated with @Controller annotation. This annotation provides Spring Boot consideration that specified class is controller. It needs for further connecting of controller class with other components. In addition, Spring Boot provides to such class HTTP requests that are sent from browser.

Controller contains methods to handle HTTP requests. Every method should have special Spring Boot annotation - @GetMapping or @PostMapping (depending on type of requests that method must recieve. Specifically, @GetMapping is a composed annotation that acts as a shortcut for @RequestMapping(method = RequestMethod.GET). It is Annotation for mapping HTTP GET requests onto specific handler methods. @PostMapping is a composed annotation that acts as a shortcut for @RequestMapping(method = RequestMethod.POST). It is annotation for mapping HTTP POST requests onto specific handler methods.

With necessary annotation should be specified which requests they should catch. When necessary request was caught, corresponding method begins to execute. Controller method needs to perform simple verification of receiving data. Then it can return view part or call methods form service part. Spring Boot also provides @Autowired annotation for automatic connection of classes from other packages. It allows to call methods from outer classes.

Having big project, there can be multiple controller classes. Usually, there are exist controller class with name MainController. In addition, every controller class can have its own name that specifies what it handles.

In this project were implemented several controller methods. Among them MainController class for main pages requests. In addition, for separate slider handling MainCardController class was implemented. This class provides methods for viewing page with sliders stored, to upload new sliders, edit existing of deleting specified slide. The same is for MainCardController class. This class provided for manipulating of cards with some information that are placed on the main page. MainCardController provides methods for viewing stored card, uploading new, editing existing and deleting specified cards. Next, CoursesDescrController provided for the same functionality of available courses handling.

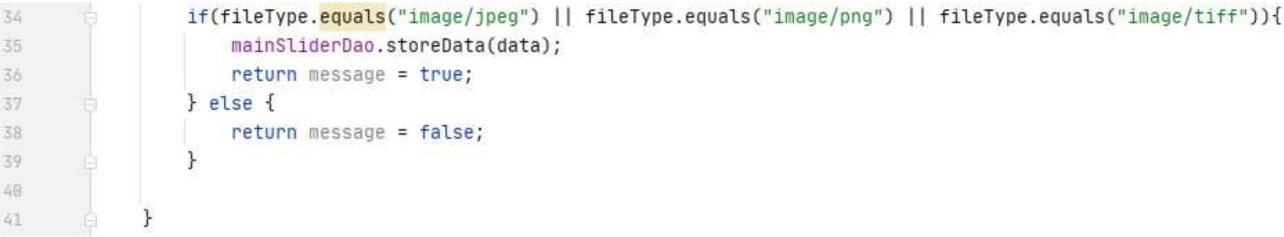
Business logic part usually located in the package named service. Service package should contain classes for processing data according to the business task.

Service class also must be specified with special annotation - @Service. Due to this annotation, Spring Boot understands which class executes business functionality.

Service package contains methods for courses list, main slider and main cards handling.

MainSliderServiceImpl class implements possibility for main page slider handling. It contains methods storeData, getAll, deletebyId and editById.

StoreData method receives all data about loaded slide. According to the business logic, it receives file (that should be an image), text for button name, text for button URL, text for slider header, text for slider text and integer number for defining time of slider showing on the page. This method validates received file. User can send, for example, document file instead of image file. Such situation is incorrect. For such validation, this method checks type of the received file. Images can be created in different types, such as jpeg, png or tiff. If received file's type is one of the image types, then storing in database procedure is executed and return Boolean message true that will be used for creation view part. In case of file type is not related to any image type, data storing procedure will not be started and Boolean message false will be returned (fig. 2.18).



```
    if(fileType.equals("image/jpeg") || fileType.equals("image/png") || fileType.equals("image/tiff")){
        mainSliderDao.storeData(data);
        return message = true;
    } else {
        return message = false;
    }

}
```

Fig. 2.18. Example of code for storing only image files

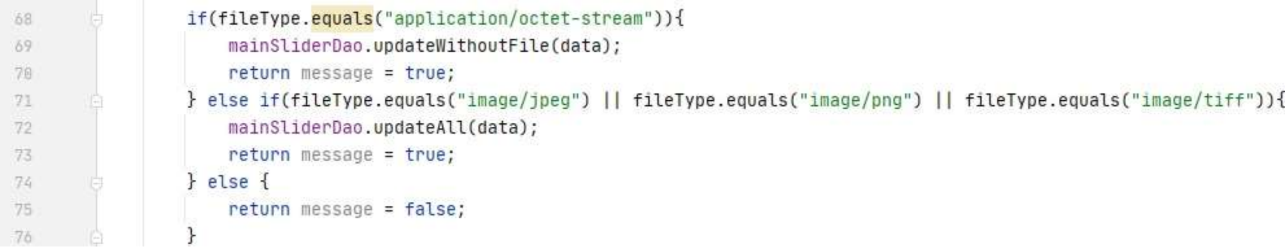
GetAll method implements procedure to retrieve all slider data from database. However, there are some issues with image data handling. So that it requires additional data manipulations.

Images are stored in Oracle database as binary large objects (BLOB). In such representation, data cannot be shown as image on browser because browser does not understand such format. Therefore, that while getting this BLOB data, it is necessary to perform data transformation into suitable format.

Received BLOB data should be coded with base64 system. Such possibility provides by the default Java library. After it was coded with base64, for showing on frontend, this data should be signed as base64 data format.

DeleteById method performs removing of the slider with specified ID. It does not return any values.

In addition, MainSliderImpl class provides editById method. This method works as storeData method, but with some differences. It checks if editing data contains new image. In the case, when new image was not provided, it stores rest data. In case, file data provided, it checks if the provided file is image and depending on check results perform storing data with return message true or simply return message false (fig. 2.19).



```
if(fileType.equals("application/octet-stream")){
    mainSliderDao.updateWithoutFile(data);
    return message = true;
} else if(fileType.equals("image/jpeg") || fileType.equals("image/png") || fileType.equals("image/tiff")){
    mainSliderDao.updateAll(data);
    return message = true;
} else {
    return message = false;
}
```

Fig. 2.19. Code to edit data depending on with image, without new image, and if provided file is an image

For manipulation with cards data, MainCardServiceImpl class is implemented. It contains methods that are similar to MainSliderServiceImpl class. However, it has some own specifics because cards have a bit another set of data.

For manipulation with courses list data, CoursesDescrServiceImpl class is implemented. It contains methods that are similar to MainSliderServiceImpl class. However, it has some own specifics because cards have a bit another set of data.

2.3.3. Communication with Database

Application should have communication with database to store and manipulate data.

Many real-world applications need to use persistent data at some point. For many apps, persistent storage is provided with big variety of mechanisms, and there are some differences in the APIs used to access these different persistent storage mechanisms. Other apps can have necessity to have an access to the data that placed on separate systems. For example, the data may reside in mainframe systems, Lightweight Directory Access Protocol (LDAP) repositories, and so forth. Another example is where data is provided by services through external systems such as business-to-business (B2B) integration systems, credit card bureau service, and so forth.

Typically, apps use shared distributed components such as entity beans to represent persistent data. An app is considered to employ bean-managed persistence (BMP) for its entity beans when these entity beans explicitly access the persistent storage-the entity bean includes code to directly access the persistent storage. An application with simpler requirements may forego using entity beans and instead use session beans or servlets to directly access the persistent storage to retrieve and modify the data. Alternatively, the application could use entity beans with container-managed persistence, and thus let the container handle the transaction and persistent details.

There is even bigger ranges with different types of persistent storage. Access mechanisms, supported APIs, and features differ between variety of types of persistent stores such as RDBMS, object-oriented databases, flat files, and so forth. Apps that have necessity to access data from a legacy or disparate system (such as a mainframe, or B2B service) are often required to use APIs that may be proprietary. Such disparate data sources provide challenges to the app and can sometimes create a direct dependency between app code and data access code. When business components-entity beans, session beans, and even presentation components like servlets and helper objects for JavaServer Pages (JSP) pages need to access a data source, they may use the appropriate API to achieve connectivity and manipulate the data source. However, including the connectivity and data access code within these components introduces a tight coupling between the components and the data source implementation. Such code dependencies in components make it difficult and tedious to migrate the application from one type of data source to another. When the data source changes, the components need to be changed to handle the new type of data source.

It is good practice to implement data access object (DAO) pattern. The data access object pattern is a structural pattern provides possibility to isolate the app or business layer from the persistence layer (often for a relational database, but it could be any other persistence mechanism) using an abstract API.

The necessity of this API is to hide from the application all the complex structures involved in performing create, read, update, and delete (CRUD) operations in the underlying storage mechanism. This allows both layers to function separately without knowing anything about each other.

The DAO provides the access mechanism offers to work with the data source. The data source could be a persistent store like an RDBMS, an external service as a B2B exchange, a repository like an LDAP database. The business component that relies on the DAO uses the simpler interface exposed by the DAO for its clients. The DAO almost hides the data source implementation details from its clients. Because the interface exposed by the DAO to clients does not change when the underlying data source implementation changes, this pattern provides possibility for the DAO to adapt to variety of storage schemes without affecting its clients or business components. Initially, the DAO acts as an adapter between the component and the data source (fig. 2.20).

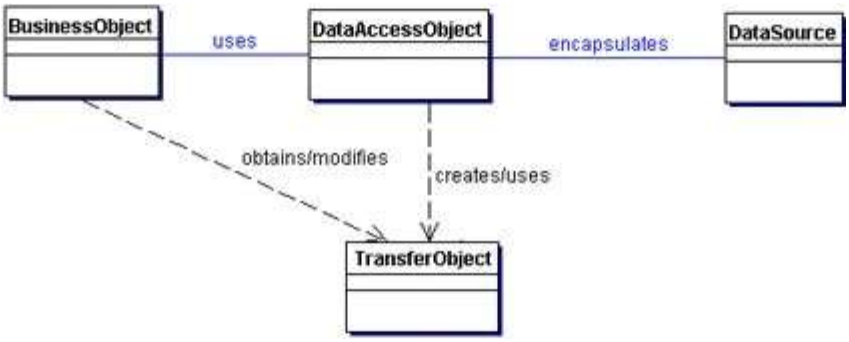


Fig. 2.20. Data Access Object scheme

The sequence diagram allows understanding the communication between the different participants in DAO pattern (fig. 2.21).

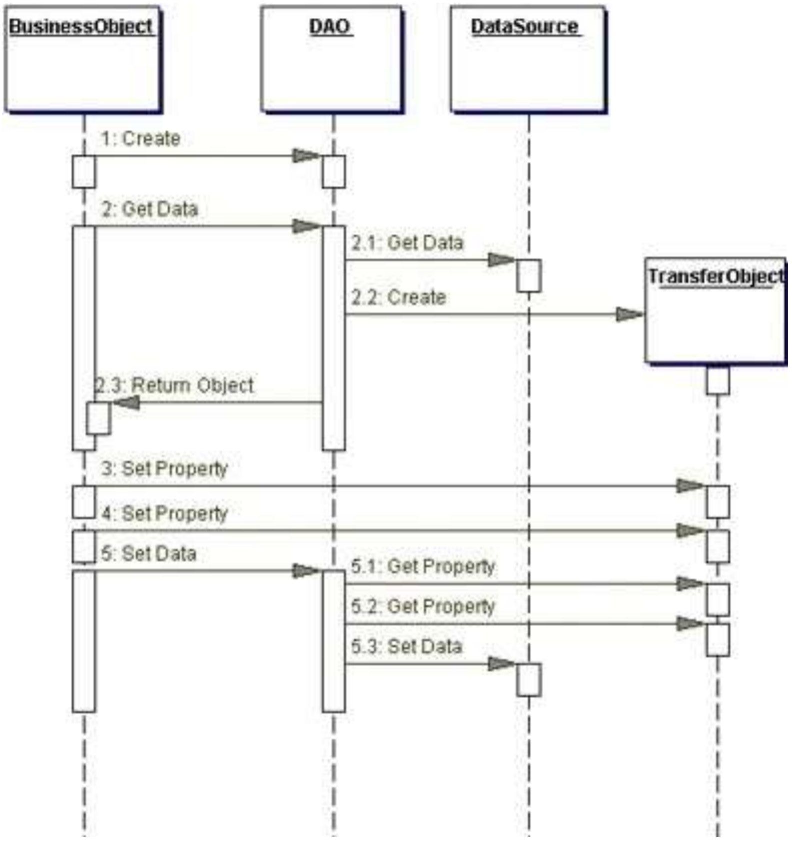


Fig. 2.21. Data Access Object sequence representation

The BusinessObject represents the data client. It is the object that needs access to the data source to process and store data. A BusinessObject can be provided as Java object.

The DataAccessObject is the main object of DAO pattern. The DataAccessObject abstracts the underlying data access implementation for the BusinessObject to provide direct access to the data source. The BusinessObject also refers data load and store operations to the DataAccessObject.

DataSource shows a data source implementation. A data source can be a database such as an RDBMS, OODBMS, XML repository, flat file system, and so forth. A data source can also be another system (legacy/mainframe), service (B2B service or credit card bureau), or some kind of repository.

TransferObject represents a Transfer Object used as a data carrier. The DataAccessObject can use a Transfer Object to return back data to the client. The DataAccessObject can also receive the data from the client in a Transfer Object to update the data in the data source [27].

To simplify creating structure for communication with database JpaRepository was introduced. It is part of Spring Boot framework. It uses Spring Data.

Spring Data is Spring-based programming model to provide data access possibility. It reduces the quantity of code necessary to work with databases. It consists of several modules. The Spring Data JPA simplifies the development of Spring apps that use JPA technology.

With Spring Data, it is necessary to define a repository interface for each domain entity in the app. A repository contains methods for performing CRUD operations, sorting and paginating data. @Repository is a marker annotation, which indicates for Spring Boot that the underlying interface is a repository. A repository is created by extending specific repository interfaces, such as CrudRepository, PagingAndSortingRepository, or JpaRepository.

Spring Data has advanced integration with Spring MVC controllers and provides dynamic query derivation from repository method names.

JpaRepository is JPA special extension of Repository. It contains the full API of CrudRepository and PagingAndSortingRepository. Therefore, it contains API for basic CRUD operations and API for pagination and sorting [28].

2.3.4. Frontend and Thymeleaf

Frontend is represented by HTML pages with JavaScript. Thymeleaf functions (tags) are written inside HTML tags. However, Thymeleaf code executes on server side and inly result as static HTML code is sent to the client.

Every page has its own HTML file. To make it not static, some JavaScript is added. Buttons activity and color changes are implemented with JavaScript (fig. 2.22, fig. 2.23).

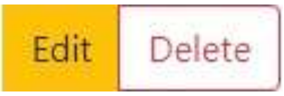


Fig. 2.22. Button position 1

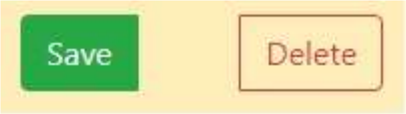
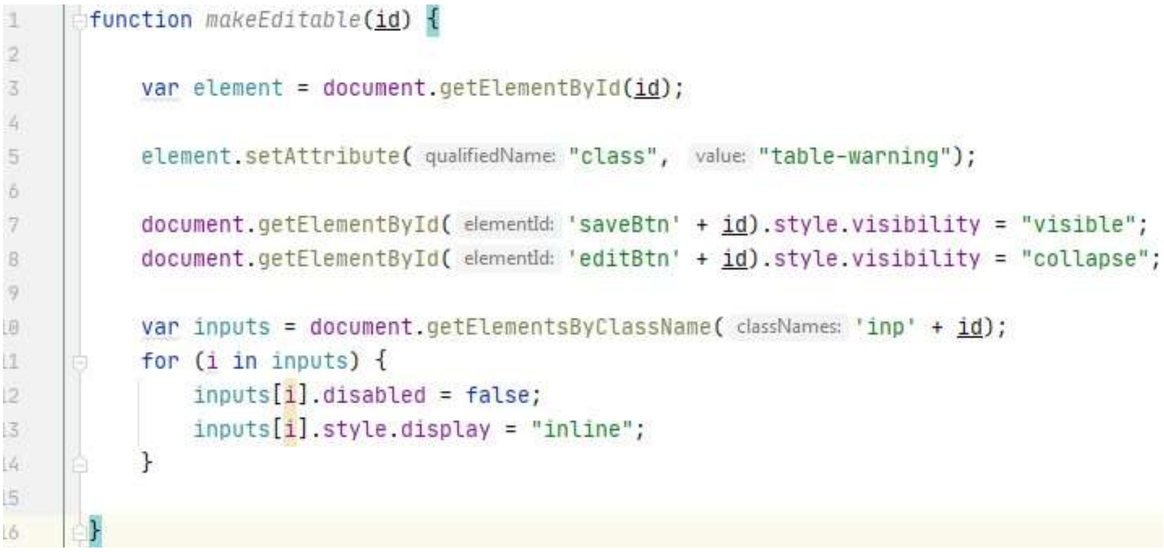


Fig. 2.23. Button position 2

At first position button Save is inactive and is not shown (blank place is shown instead). At second button position, button Edit is inactive and Save is active instead. Such situation is implemented with the help of JavaScript (fig. 2.24).

According to widely used practice, JavaScript code should be separated in distinct file.



```
function makeEditable(id) {

    var element = document.getElementById(id);

    element.setAttribute( qualifiedName: "class",  value: "table-warning");

    document.getElementById( elementId: 'saveBtn' + id).style.visibility = "visible";
    document.getElementById( elementId: 'editBtn' + id).style.visibility = "collapse";

    var inputs = document.getElementsByClassName( classNames: 'inp' + id);
    for (i in inputs) {
        inputs[i].disabled = false;
        inputs[i].style.display = "inline";
    }

}
```

Fig. 2.24. Block of JavaScript code to change button appearance

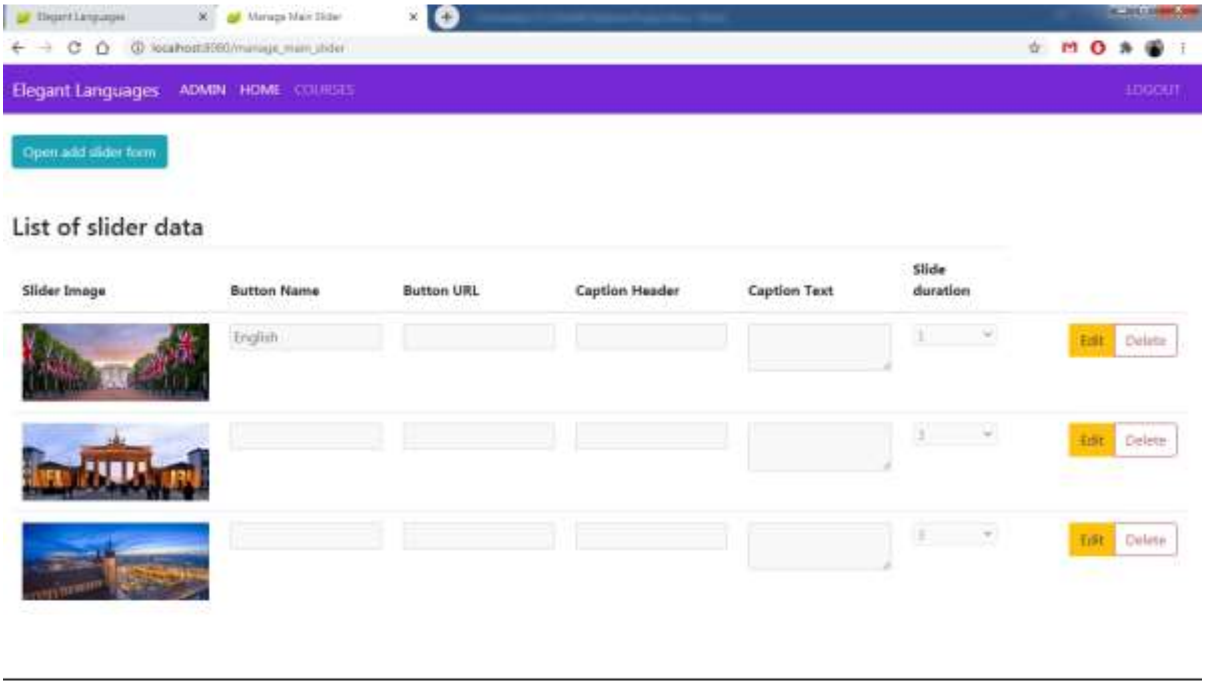
This code is called from HTML page with these buttons defined. At the same time, not only button appearance is changed but also fields become editable.

Usually, to make page layout more user friendly uses CSS. In this case Bootstrap framework was used.

Deploying a project is not simple. It requires hours and hours of work and the programming skills needed are intense. One single mistake can destroy the whole project, so the quantity of stress and pressure that is put on the developer is enormous. As a framework, what Bootstrap does is to make the process of development simpler, by keeping the code consistent and of high quality. Human mistakes are normal and having a well-tested and proven framework to build on is extremely convenient.

Bootstrap is the most popular CSS Framework for developing responsive and mobile-first websites.

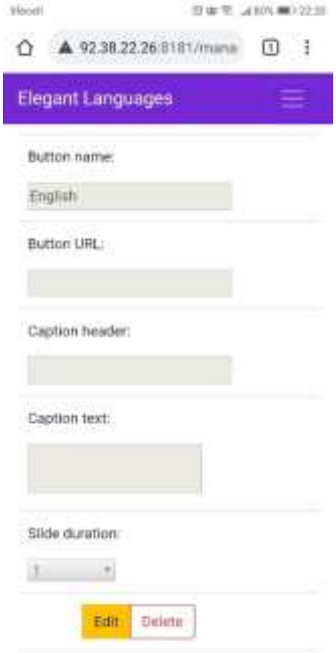
It provides classes that are easy to write. Every class has own name that helps to understand its usage. In addition, Bootstrap was designed with flexibility in mind. So that, this web application is designed to be suitable for devices with both big and small screens like on smartphones (fig. 2.25, fig. 2.26).



List of slider data

Slider Image	Button Name	Button URL	Caption Header	Caption Text	Slide duration	
	English				1	Edit Delete
					1	Edit Delete
					1	Edit Delete

Fig. 2.25. View on big screen



Button name: English

Button URL:

Caption header:

Caption text:

Slide duration: 1

Edit Delete

Fig. 2.26. View on smartphone screen

To make frontend layout dynamic, Thymeleaf is used. It is executed on server side, but is written in HTML pages that refers to frontend side.

Thymeleaf is not web framework. It is a template engine. Template engines are very important in web frameworks, and are one of its most important elements as they are in charge of producing the user interface or view layer (usually in XHTML or HTML form). Spring MVC, Struts or Apache Wicket are examples of web frameworks, whereas JSP, Velocity or FreeMarker are examples of template engines.

Thymeleaf templates can be:

1) HTML;

2) XML;

3) TEXT;

4) JAVASCRIPT;

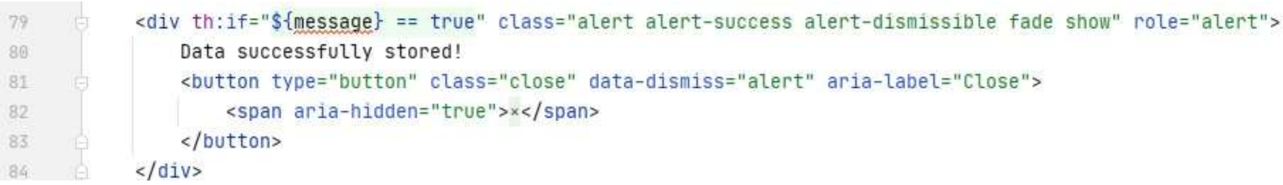
5) CSS.

Thymeleaf cab be used as a whole substitute for JSP.

Both Velocity and FreeMarker templates engines are terrific pieces of software, but they approach the templating problem with a philosophy quite different to that of Thymeleaf.

Thymeleaf makes a strong emphasis on natural templating — making possible for templates to be working prototypes, which the other two do not allow —, and its syntax tries to be (arguably) cleaner and more in tune with the current trends in web development. In addition, from an architectural vision, both Velocity and FreeMarker work as sequential text processors whereas Thymeleaf is based on markup parsing techniques. This allows Thymeleaf to take advantage of interesting features specific to markup-based environments, especially the web.

Although Thymeleaf (especially its Standard dialects) offers big variety of tools that are especially suitable for web development, it can be used for executing non-web HTML or XML documents (data XML, for example) or other types of templates that are not meant for being sent via HTTP (for example, text/HTML email content) (fig. 2.27).



```
<div th:if="${message} == true" class="alert alert-success alert-dismissible fade show" role="alert">
    Data successfully stored!
    <button type="button" class="close" data-dismiss="alert" aria-label="Close">
        <span aria-hidden="true">×</span>
    </button>
</div>
```

Fig. 2.27. Example of Thymeleaf usage inside HTML code

There are standard expression syntax:

1) ${...} : Variable expressions;

2) *{...} : Selection expressions;

3) #{...} : Message expressions, allows to retrieve locale-specific messages from external sources;

4) @{...} : Link (URL) expressions;

5) ~{...} : Fragment expressions.

Conclusions of the Second Part

Web application development requires a wide range of tools. At first, it is necessary to know software tools providing development possibilities. Integrated development environments should support technologies that are going to be used in development. Next, database management service must be installed to have possibility of database creation. Process of preparing for development may require necessary settings. This stage may occur issues.

To develop web application, programming pattern should be chosen. Then, technologies, programming language and frameworks must be chosen and adjusted.

Programming pattern provides project structure. It is important to choose pattern according task that application will implement. In web development MVC pattern is widely used. MVC stands for Model-View-Controller. Main feature of this pattern is to separate business logic of web application from other parts. Such separation simplifies modifying of business logic part.

Frameworks help to decrease time that was spend on development. They have predefined blocks of code, so that it is not necessary to implement them manually.

Databases a main storage for application. It provides possibility to store values even in case of application stops. In addition, databases organize data more useful than it can be made inside application.

Structure of the project plays important role. It is not only simplify navigation through the project, but also provides better support for external frameworks because they can use standard structure.

PART 3
WEB APPLICATION FUNCTIONALITY AND RELIABILITY

3.1. Web Application Functionality

Web application provides some possibilities. Possibilities determines depending on the reason for what application was developed. Functionality may include both business logic realizations and features for better user experience.

Web application provides possibility to administrator to perform some configurations on main page (fig. 3.1) and manage courses page content.

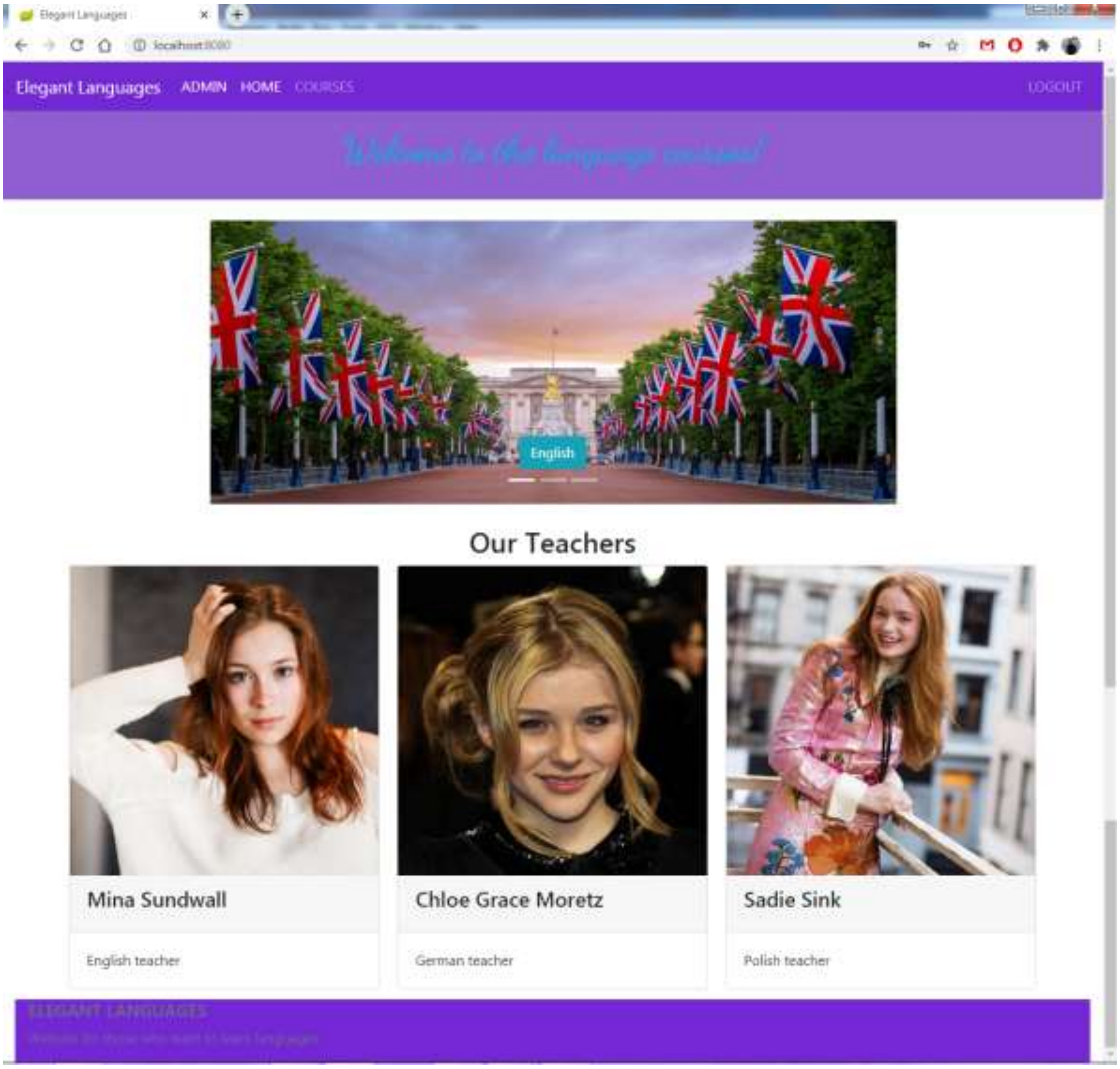


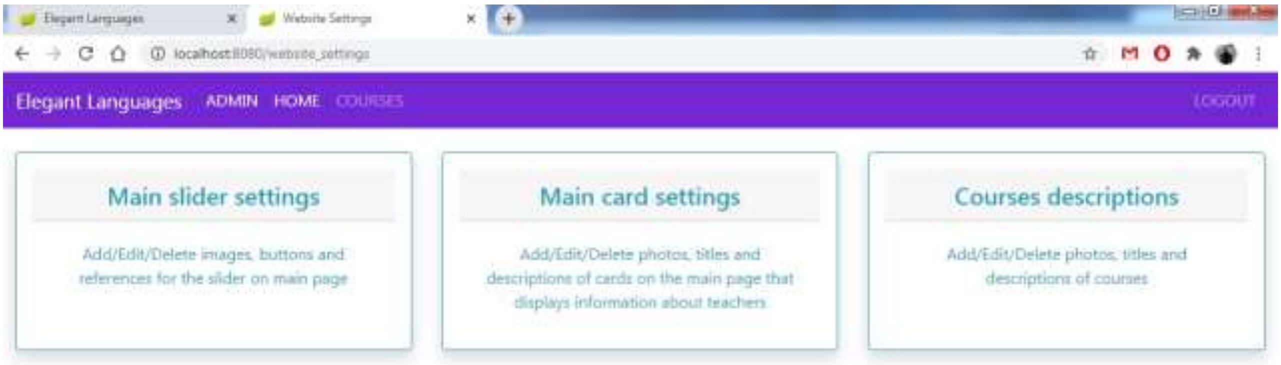
Fig. 3.1. Main page of web application

Main page consist of two main elements: slider and cards with teacher’s info.

Slider shows images that change each other automatically or with mouse click. In case when mouse is on the slider, it stops. This slider can have different layouts: with or without button (with link to another page), with or without bigger and smaller description text. In addition, slider can have different layout duration. Every slider can have its own layout.

Cards represents info about teachers. They can be with or without image, with or without bold text (for names), with or without plain text (for descriptions) and with or without button with link. Every card can have different layout.

Layout of slider and cards are not hardcoded. Instead, there are provided functionality for website administrator to manage them. Pages to control slider and cards are accessible via website settings page (fig. 3.2).



Main slider settings

Add/Edit/Delete images, buttons and references for the slider on main page

Main card settings

Add/Edit/Delete photos, titles and descriptions of cards on the main page that displays information about teachers

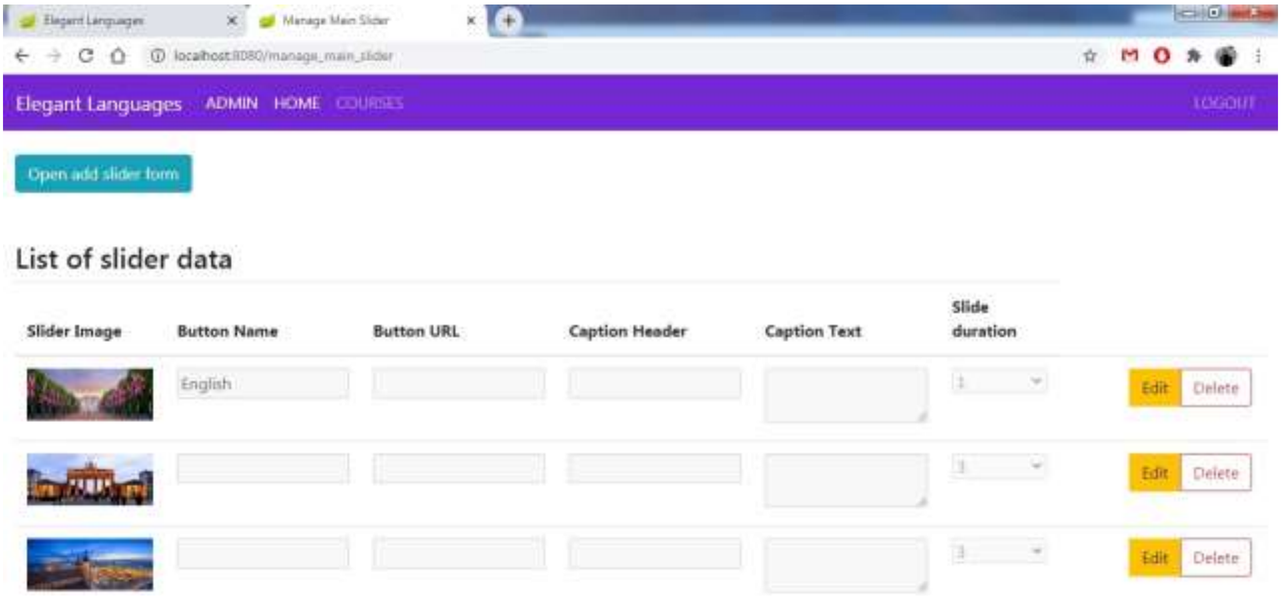
Courses descriptions

Add/Edit/Delete photos, titles and descriptions of courses

Fig. 3.2. Website settings page of web application

This page provide buttons to access settings that are related to configuration of main slider, main card and courses description.

To access slider settings it is necessary to press Main slider settings button. After that, special page with necessary fields and parameters for slider manipulation will be opened (fig. 3.3). This page combines two purposes. It provides possibility to upload new images with related info for slider and to review already uploaded sliders information.

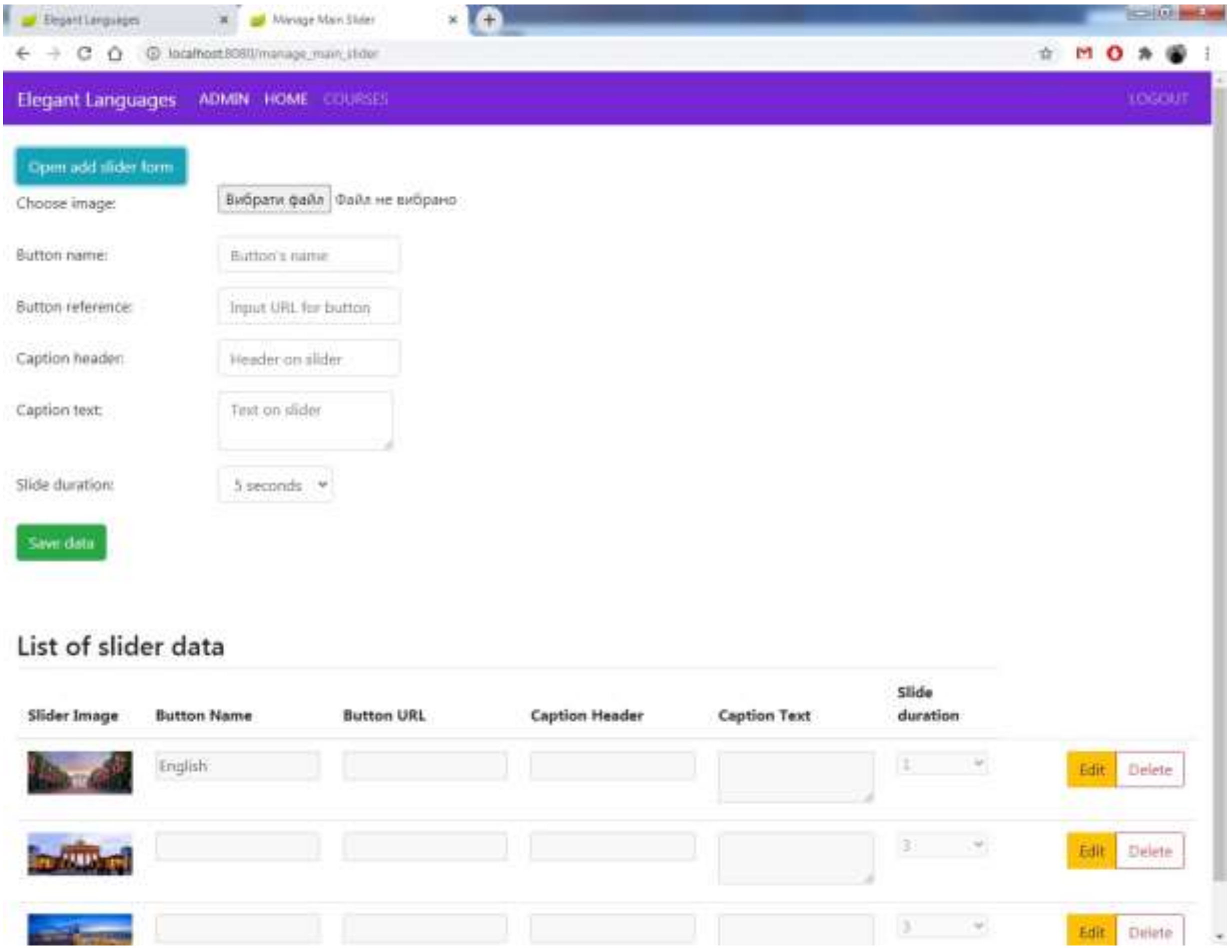


List of slider data

Slider Image	Button Name	Button URL	Caption Header	Caption Text	Slide duration	
	English				1	Edit Delete
					3	Edit Delete
					3	Edit Delete

Fig. 3.3. Manage Main Slider page

Although Manage Main Slider page allows to upload slider data, the necessary form for data uploading is hided. There are only Open add slider form button, after pressing which, form for data is opened (fig. 3.4).



Choose image:	Вибрати файл Файл не вибрано
Button name:	Button's name
Button reference:	Input URL for button
Caption header:	Header on slider
Caption text:	Text on slider
Slide duration:	5 seconds

List of slider data

Slider Image	Button Name	Button URL	Caption Header	Caption Text	Slide duration	
	English				1	Edit Delete
					3	Edit Delete
					3	Edit Delete

Fig. 3.4. Manage main slider page with opened form

Repeated pressing on Open add slider form button will occur form closing. First row in form is for image choosing. There are no size restrictions. Next row is for button name – text that will appear on the created button (fig. 3.5). Button reference stands for link that will be assigned to the button.

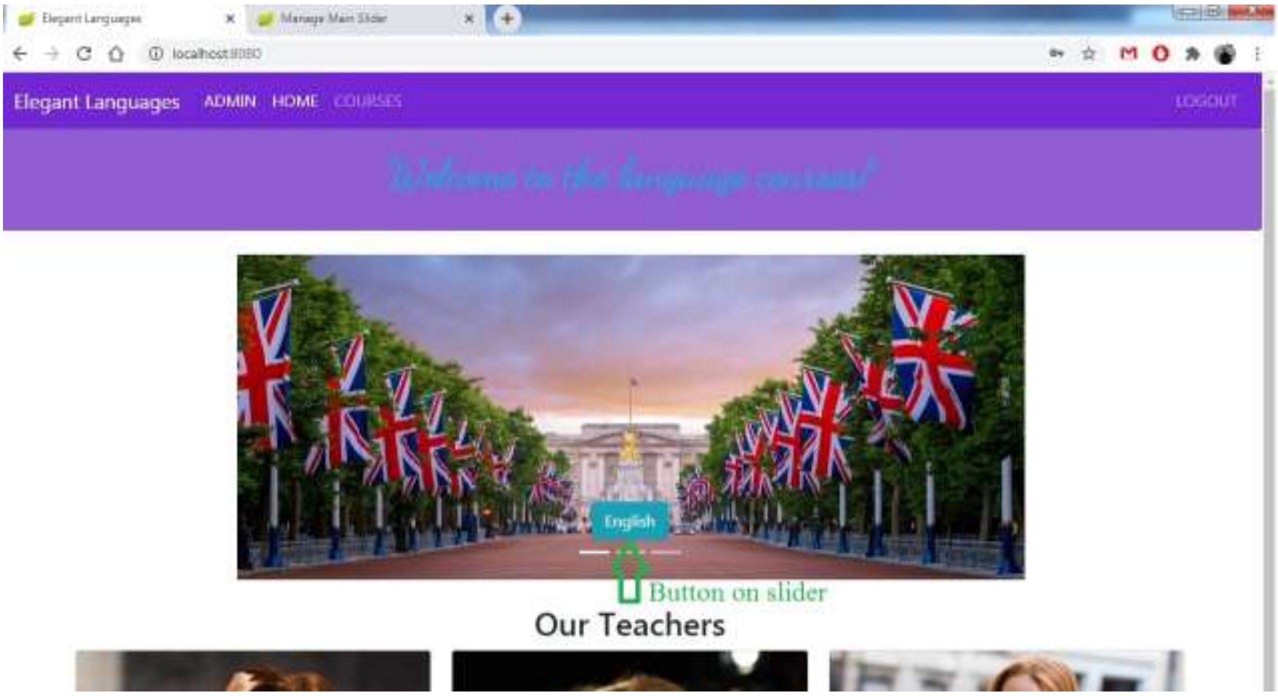


Fig. 3.5. Main page slider button (signed by arrow)

Caption header refers to text to write over slider with header font. Caption text refers to text write over slider after header text (fig. 3.6).

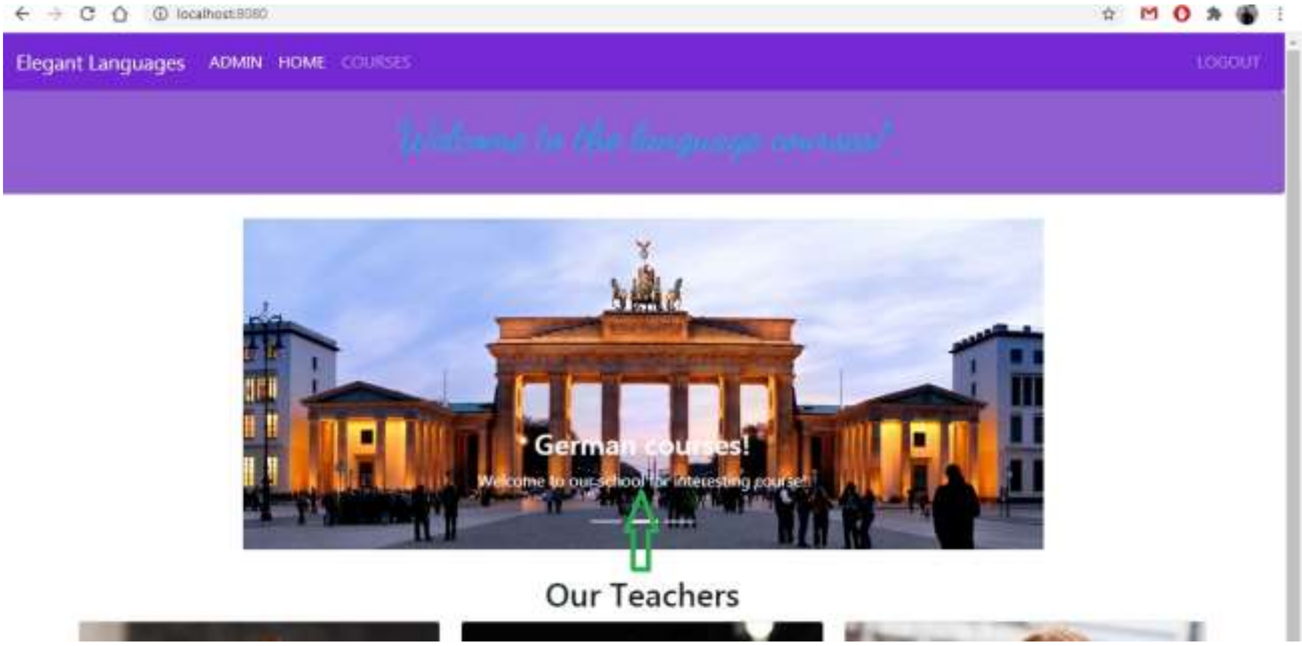
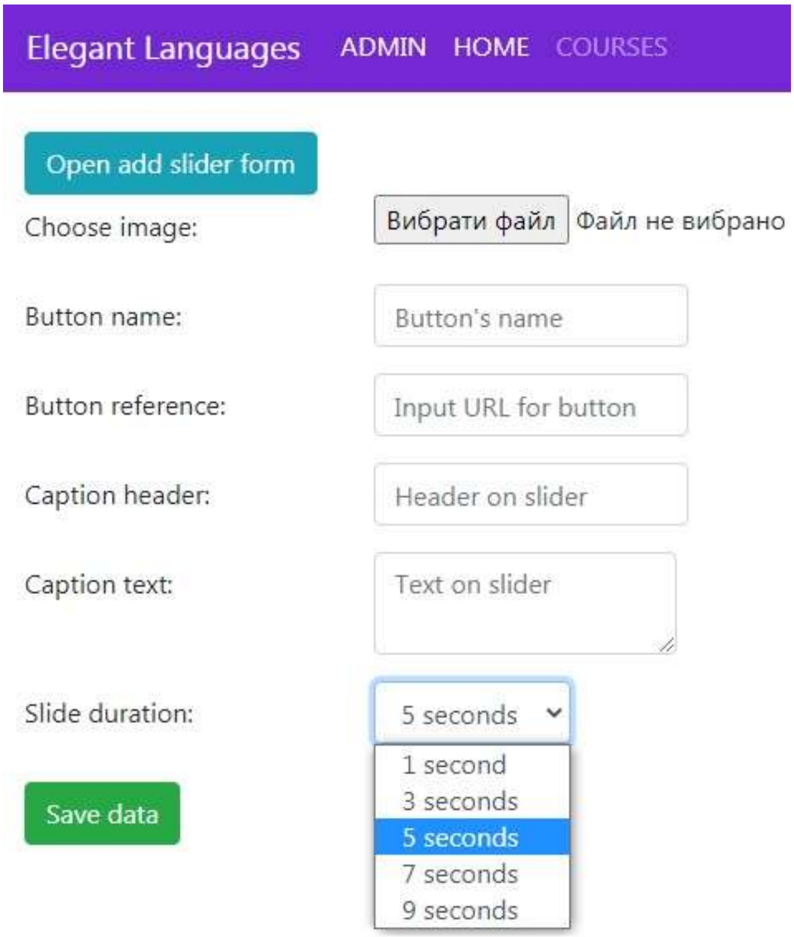


Fig. 3.6. Main page slider with text instead of button (signed by arrow)

Last row in form refers to time for what slider image will be shown on the page. It is drop-down menu with such variants: 1, 3, 5, 7 and 9 seconds (fig. 3.7).



Choose image: Файл не вибрано

Button name: Button's name

Button reference: Input URL for button

Caption header: Header on slider

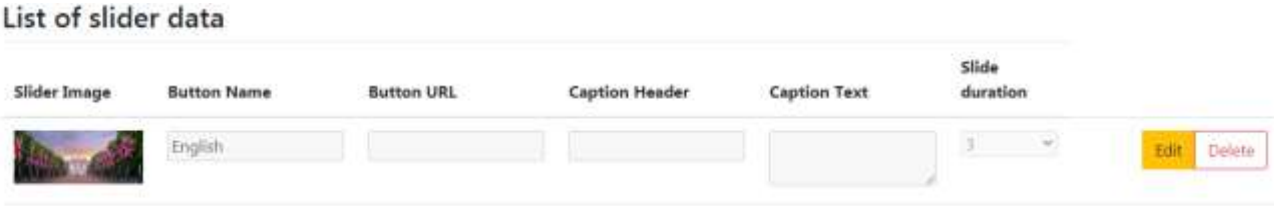
Caption text: Text on slider

Slide duration: 5 seconds

Fig. 3.7. Add slider form with drop-down menu for slide duration

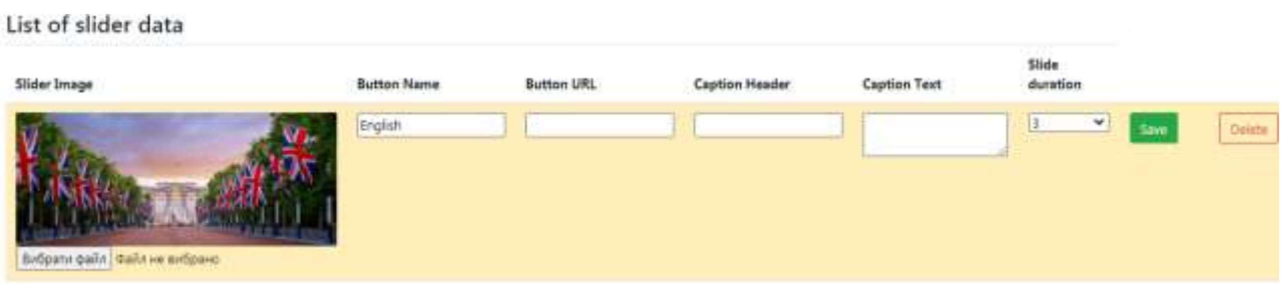
Section that displays list of slider data also has its features. Every row of the table displays fields for every data item. They can be empty if there are no data. They appear in disable mode (fig. 3.8). To edit fields it is necessary to press edit button (fig. 3.9). It useful to prevent unnecessary typing into fields when there are no necessity for performing edition.

Save button will appear after Edit button will be pressed.


List of slider data

Slider Image	Button Name	Button URL	Caption Header	Caption Text	Slide duration	
	English				3	Edit Delete

Fig. 3.8. Row of slider data list with disabled fields



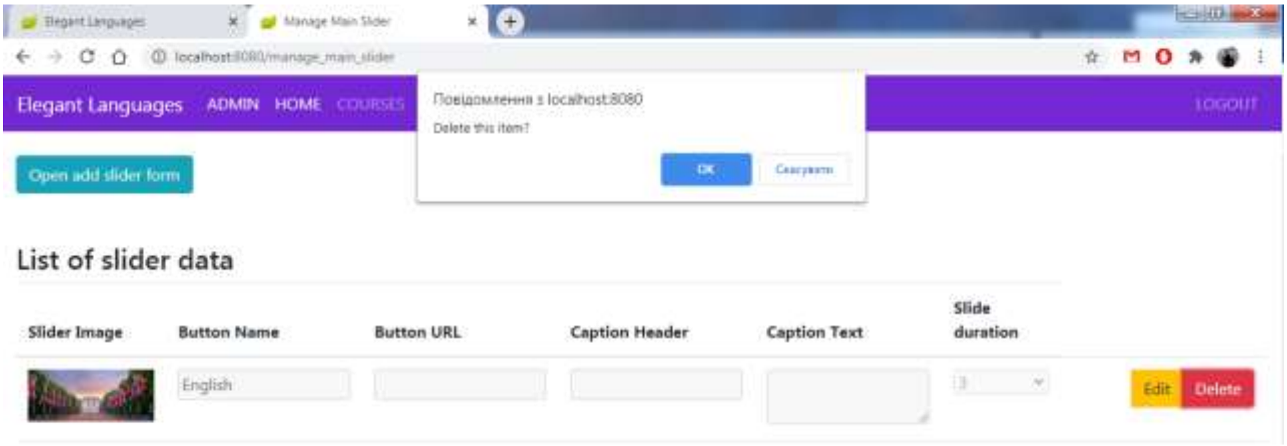
List of slider data

Slider Image	Button Name	Button URL	Caption Header	Caption Text	Slide duration
Вибрати файл Файл не вибрано	English				3

Fig. 3.9. Row of slider data list with enabled fields for edition

Edition possible for any field including new image uploading. It is possible to edit some field or fields without uploading new image or with uploading at the same time.

It is possible not only edit fields, but also delete slider with its data. To prevent unnecessary deletion, attention pop-up message appears after Delete button was pressed (fig. 3.10).



Повідомлення з localhost:8080

Delete this item?

List of slider data

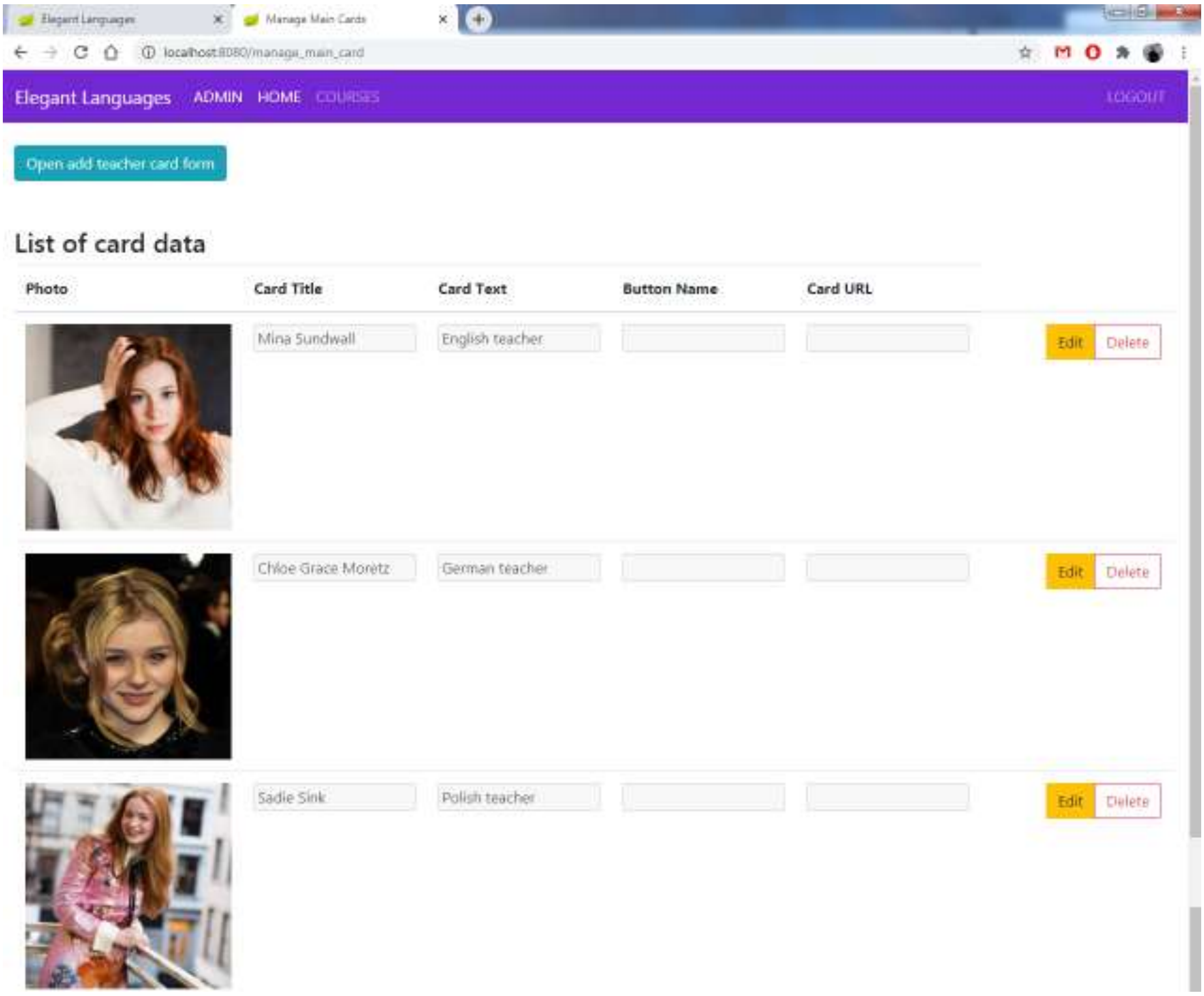
Slider Image	Button Name	Button URL	Caption Header	Caption Text	Slide duration
	English				3

Fig. 3.10. Manage main slider page after Delete button was pressed

Pressing cancel button in pop-up message will cancel deletion action. If OK is pressed, necessary data will be sent to the backend to perform data deletion.

Manage main slider page is effectively suitable for both big and small screens. On previous figures, it was shown layout for big screens. On smartphone size screen this list table will transform columns into rows for better layout.

To access card settings it is necessary to press Main card settings button. After that, special page with necessary fields and parameters for cards manipulation will be opened. This page combines two purposes. It provides possibility to upload new images with related info for cards and to review already uploaded sliders information. In addition, it is possible to edit existing card data or delete card (fig. 3.11).



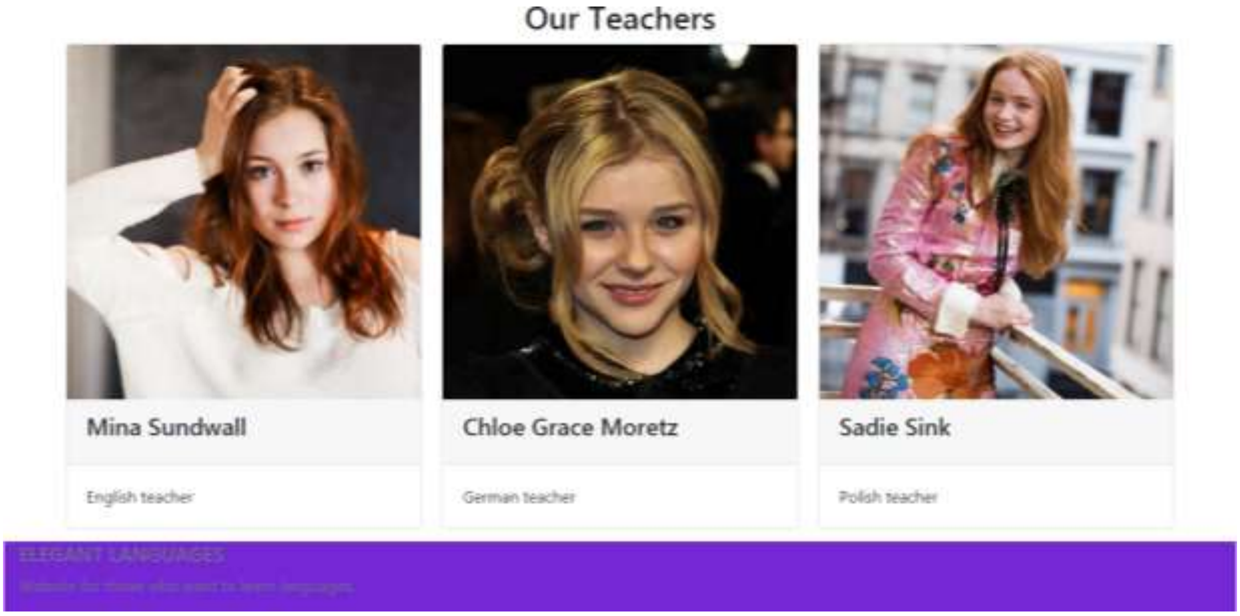
List of card data

Photo	Card Title	Card Text	Button Name	Card URL	
	Mina Sundwall	English teacher			Edit Delete
	Chloe Grace Moretz	German teacher			Edit Delete
	Sadie Sink	Polish teacher			Edit Delete

Fig. 3.12. Manage main cards page layout

These cards refer to the special rectangular blocks on main page to display teacher’s information (fig. 3.13). On the top image is placed. Usually, it should be photo of the teacher. Next, lower the photo, teacher’s name is displayed. One lower, description text is printed. Lower description, button with link can be added.

All these data is set on manage main card page. Photo is obvious for downloading; other data can be downloaded depending on necessity.



Our Teachers

Mina Sundwall

English teacher

Chloe Grace Moretz

German teacher

Sadie Sink

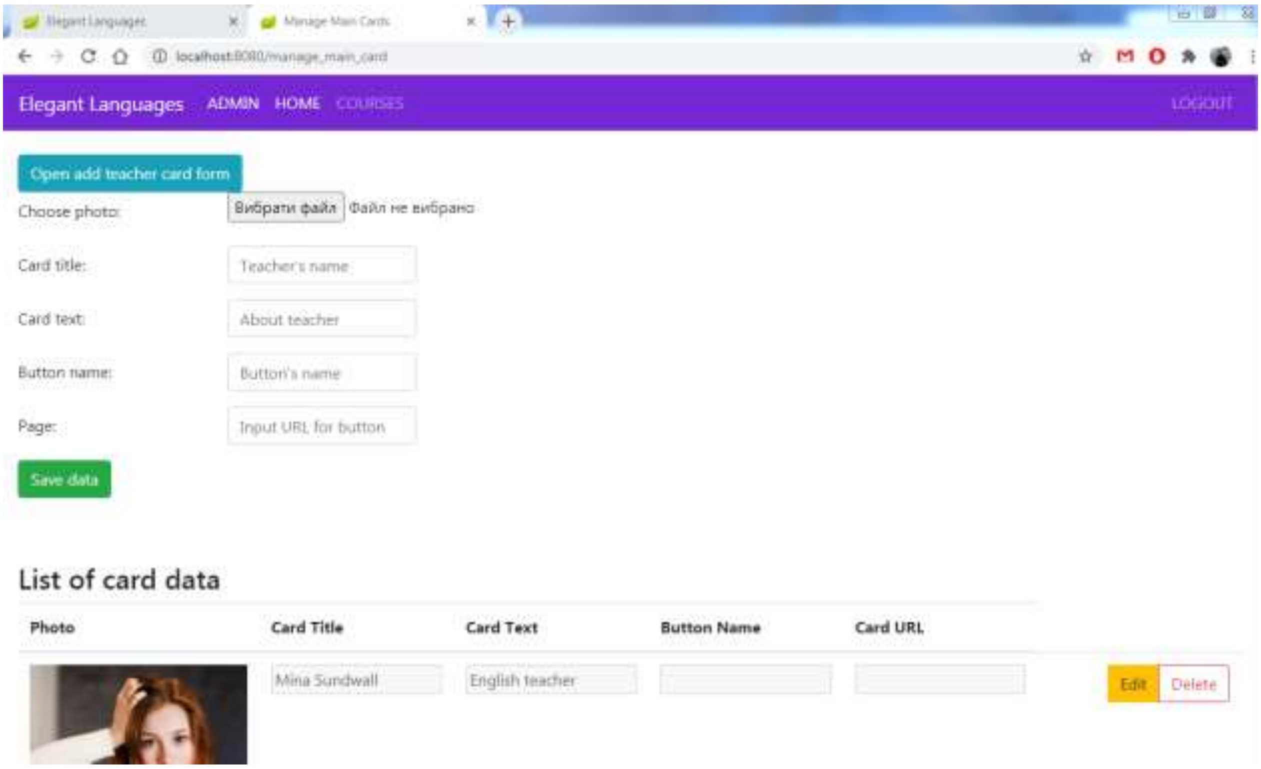
Polish teacher

ELEGANT LANGUAGES

Website for those who want to learn languages

Fig. 3.13. Cards on main page

Although manage main card page allows to upload card data, the necessary form for data uploading is hided. There are only Open add teacher card form button, after pressing which, form for data is opened (fig. 3.14).



Choose photo: Файл не вибрано

Card title: Teacher's name

Card text: About teacher

Button name: Button's name

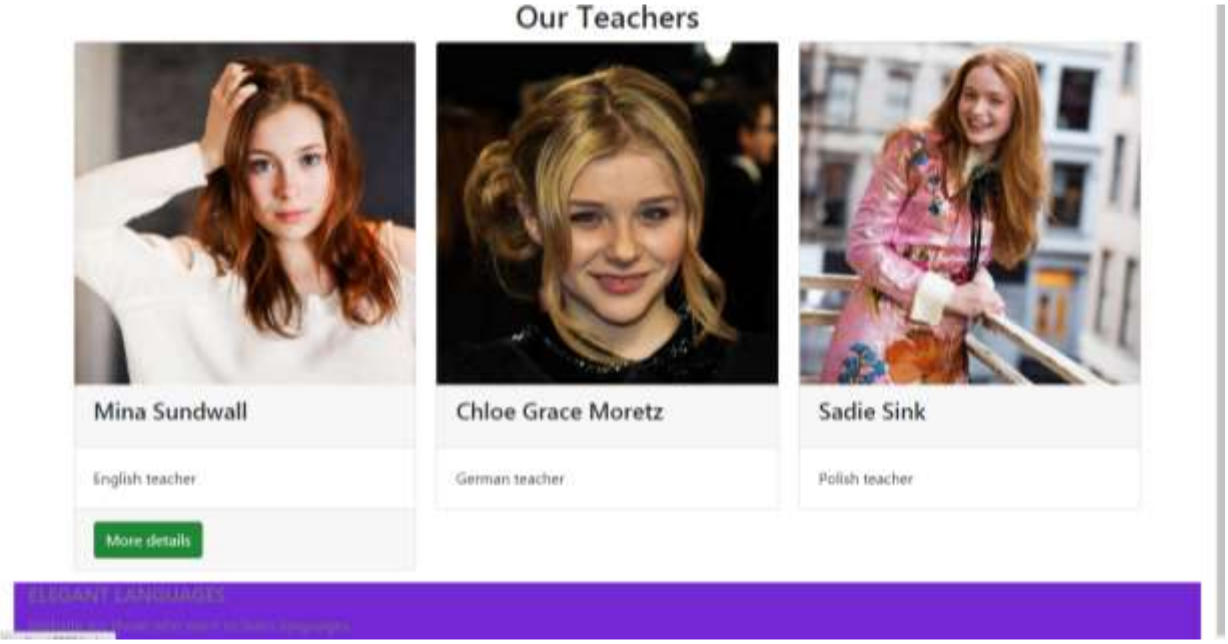
Page: Input URL for button

List of card data

Photo	Card Title	Card Text	Button Name	Card URL	
	Mina Sundwall	English teacher			Edit Delete

Fig. 3.14. Manage main cards page with opened form

Repeated pressing on Open add teacher card form button will occur form closing. First row in form is for image choosing. There are no size restrictions. Second row stands for card title. It will display as teacher’s name on main page. After that, card text row follows. Card text stands for teacher’s description. Next row is for button name – text that will appear on the created button below teacher’s description (fig. 3.15). Page row stands for link that will be assigned to the button.


Our Teachers

Mina Sundwall

English teacher

More details

Chloe Grace Moretz

German teacher

Sadie Sink

Polish teacher

ELEGANT LANGUAGES

Fig. 3.15. Card on main page with button appeared

Section that displays list of card data also has its features. Every row of the table displays fields for every data item. They can be empty if there are no data. They appear in disable mode (fig. 3.16). To edit fields it is necessary to press edit button (fig. 3.17). It useful to prevent unnecessary typing into fields when there are no necessity for performing edition. Save button will appear after Edit button will be pressed.

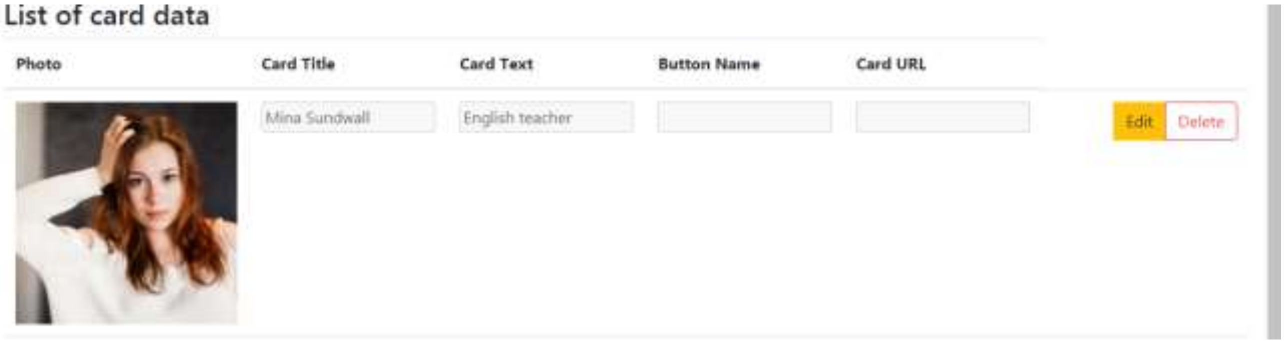
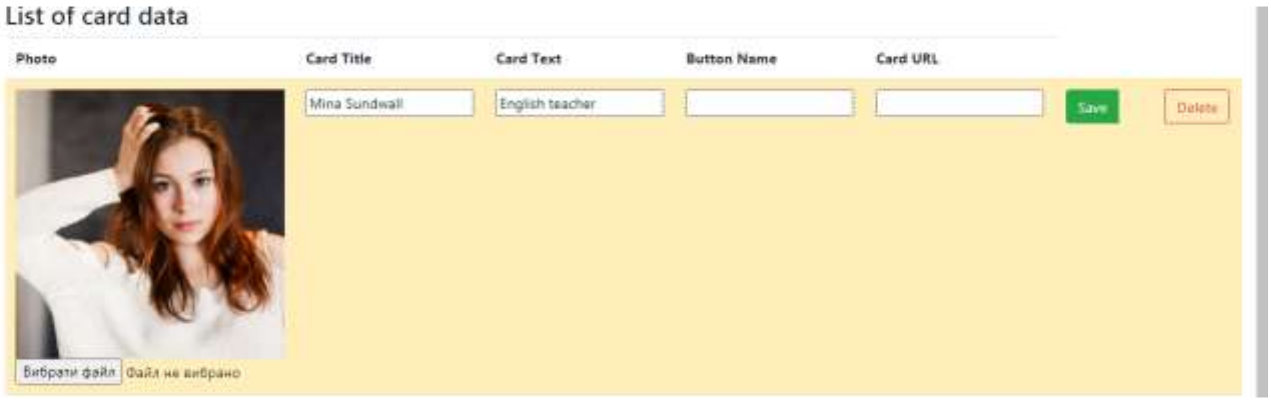

List of card data

Photo	Card Title	Card Text	Button Name	Card URL	
	Mina Sundwall	English teacher			Edit Delete

Fig. 3.16. Row with card data while fields are disabled



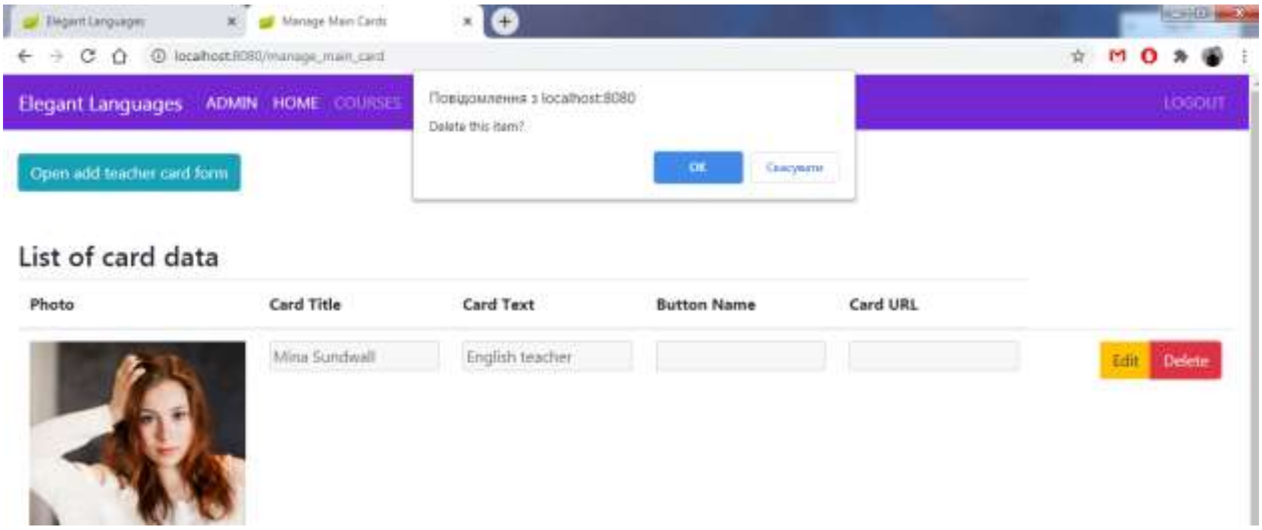
List of card data

Photo	Card Title	Card Text	Button Name	Card URL		
Вибрати файл Файл не вибрано	Mina Sundwall	English teacher			Save	Delete

Fig. 3.17. Row with card data while fields enabled for edition

Edition possible for any field including new image uploading. It is possible to edit some field or fields without uploading new image or with uploading at the same time.

It is possible not only edit fields, but also delete card with its data. To prevent unnecessary deletion, attention pop-up message appears after Delete button was pressed (fig. 3.18).



Повідомлення з localhost:8080

Delete this item?

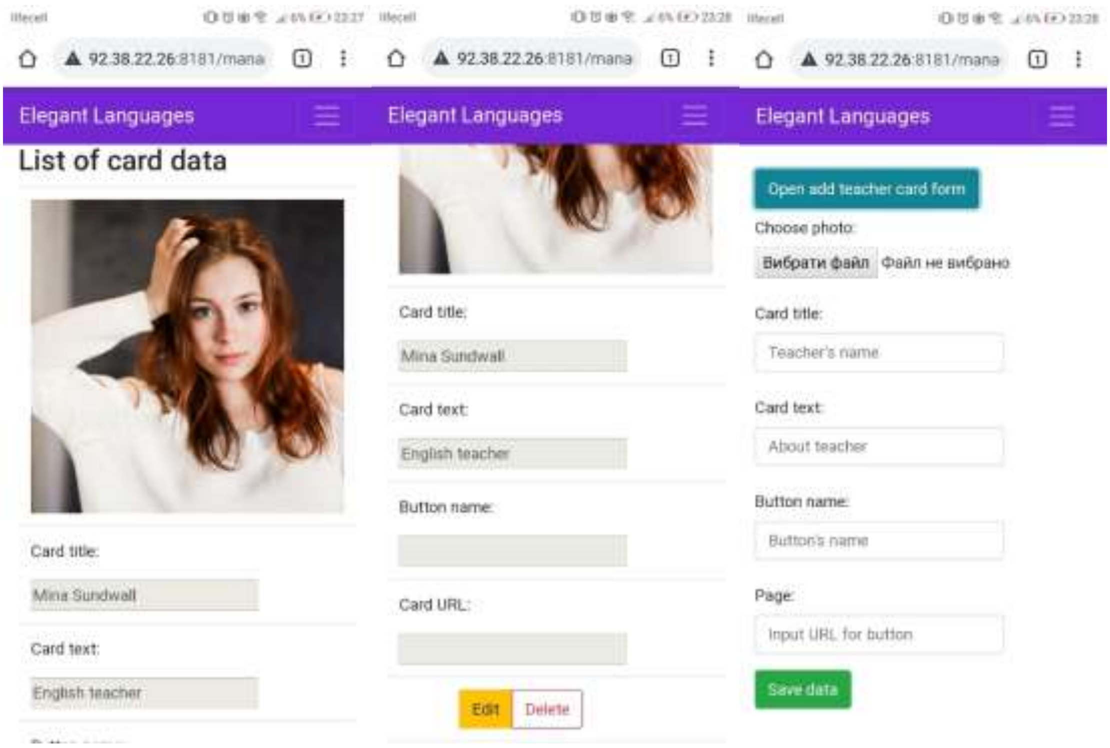
List of card data

Photo	Card Title	Card Text	Button Name	Card URL	
	Mina Sundwall	English teacher			Edit Delete

Fig. 3.18. Manage main cards page with pop-up message for deletion suggesting

Pressing cancel button in pop-up message will cancel deletion action. If OK is pressed, necessary data will be sent to the backend to perform data deletion (the same as in case with manage main slider page).

Manage main cards page is effectively suitable for both big and small screens. On previous figures, it was shown layout for big screens. On smartphone size screen this list table will transform columns into rows for better layout (fig. 3.19).



List of card data

Card title: Mina Sundwall

Card text: English teacher

Card title: Mina Sundwall

Card text: English teacher

Button name:

Card URL:

Choose photo: Вибрати файл Файл не вибрано

Card title: Teacher's name

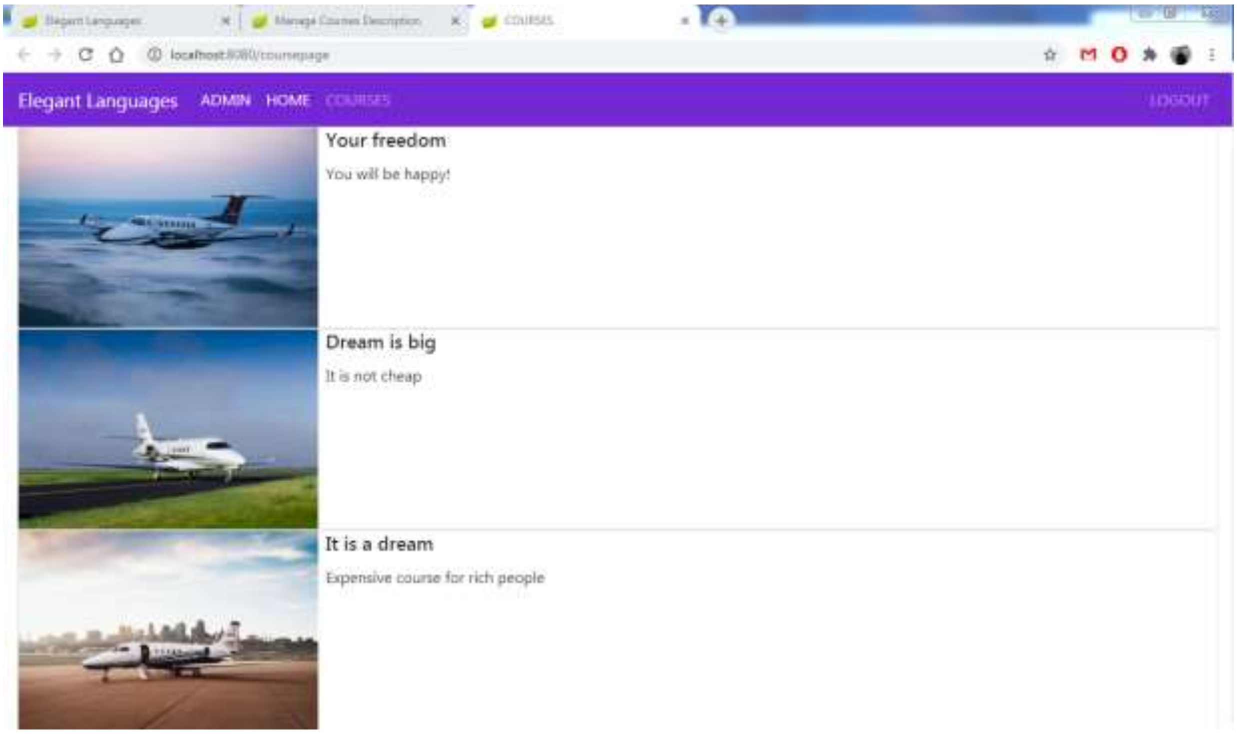
Card text: About teacher

Button name: Button's name

Page: Input URL for button

Fig. 3.19. Manage main cards page layout on smartphone screen. Left and middle image shows list data layout, right image shows upload form layout

Website contains separate page for displaying courses available. This page can be accessed by anyone and is necessary only for displaying courses (fig. 3.20).



Your freedom

You will be happy!

Dream is big

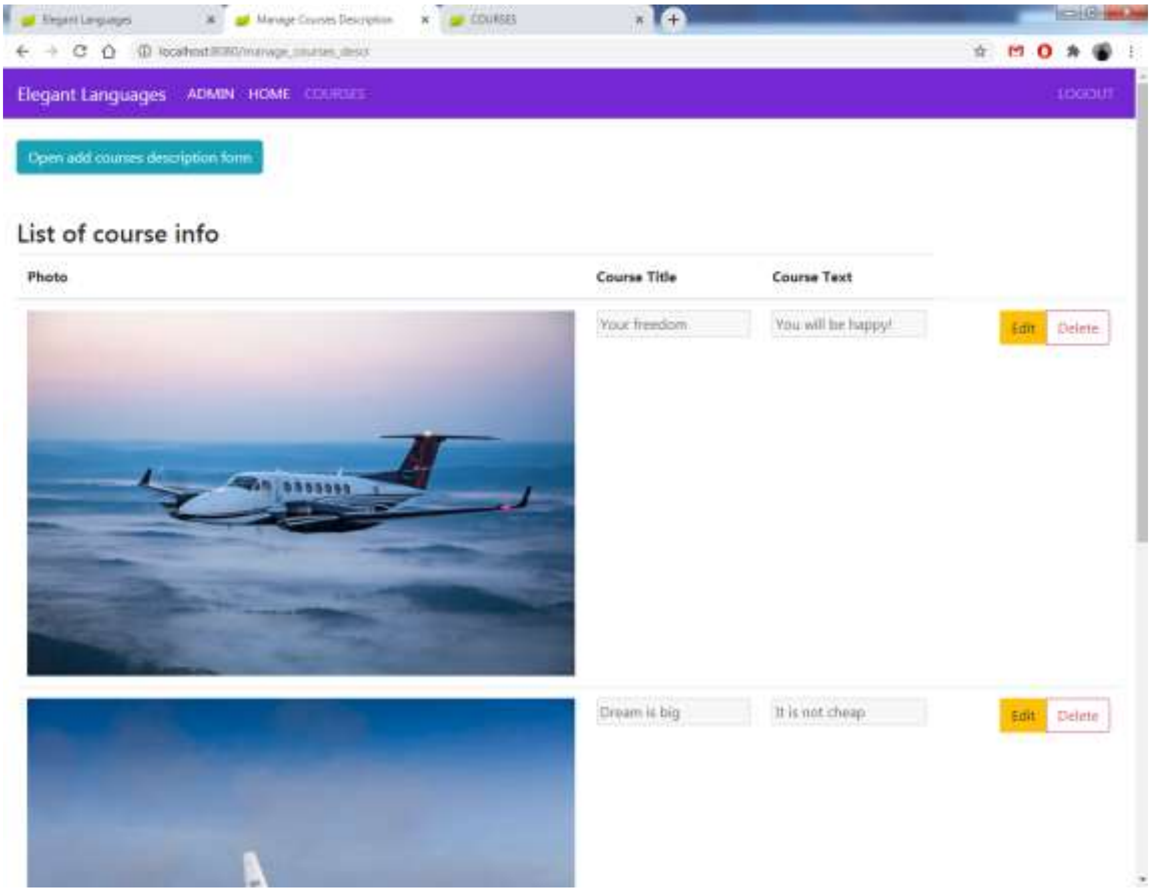
It is not cheap

It is a dream

Expensive course for rich people

Fig. 3.20. Course page with courses available

This page can be filled with image, course title and course description. Every course data can be managed via Manage courses description page by administrator (fig. 3.21).

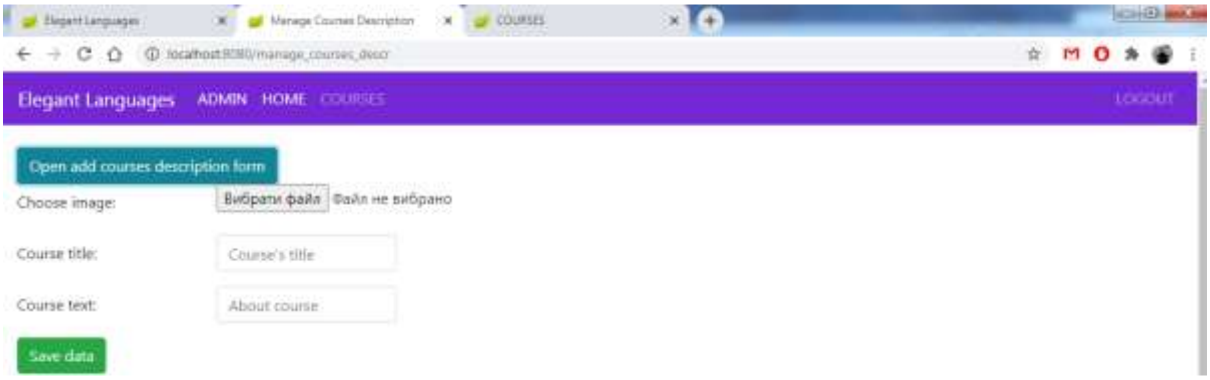

Elegant Languages
Manage Courses Description
COURSES
localhost:8080/manage_courses_desc
Elegant Languages
ADMIN
HOME
COURSES
LOGOUT
Open add courses description form

List of course info

Photo	Course Title	Course Text	
	Your freedom	You will be happy!	Edit Delete
	Dream is big	It is not cheap	Edit Delete

Fig. 3.21. Manage courses page layout

These page layouts two sections – for uploading new data and for reviewing, editing or deleting existing data. To upload new course data Open add courses description form button should be pressed (fig. 3.22).


Elegant Languages
Manage Courses Description
COURSES
localhost:8080/manage_courses_desc
Elegant Languages
ADMIN
HOME
COURSES
LOGOUT
Open add courses description form
Вибрати файл
Save data

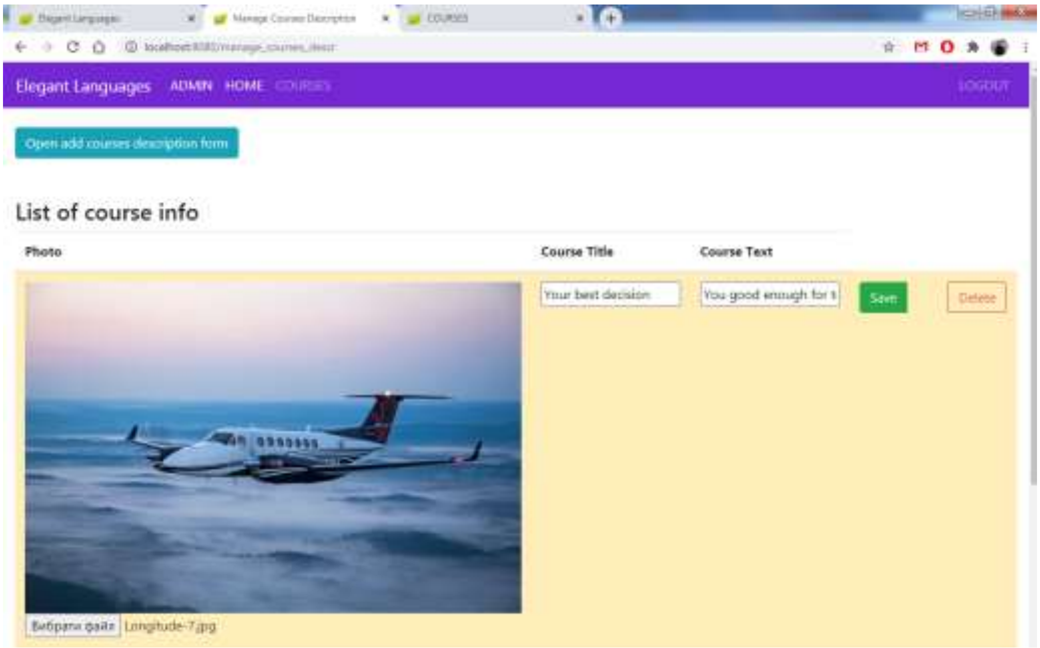
Choose image: Вибрати файл Файл не вибрано

Course title: Course's title

Course text: About course

Fig. 3.22. Add courses data form opened

Editing possibilities are provided in the same way as for cards and sliders (fig. 3.23).

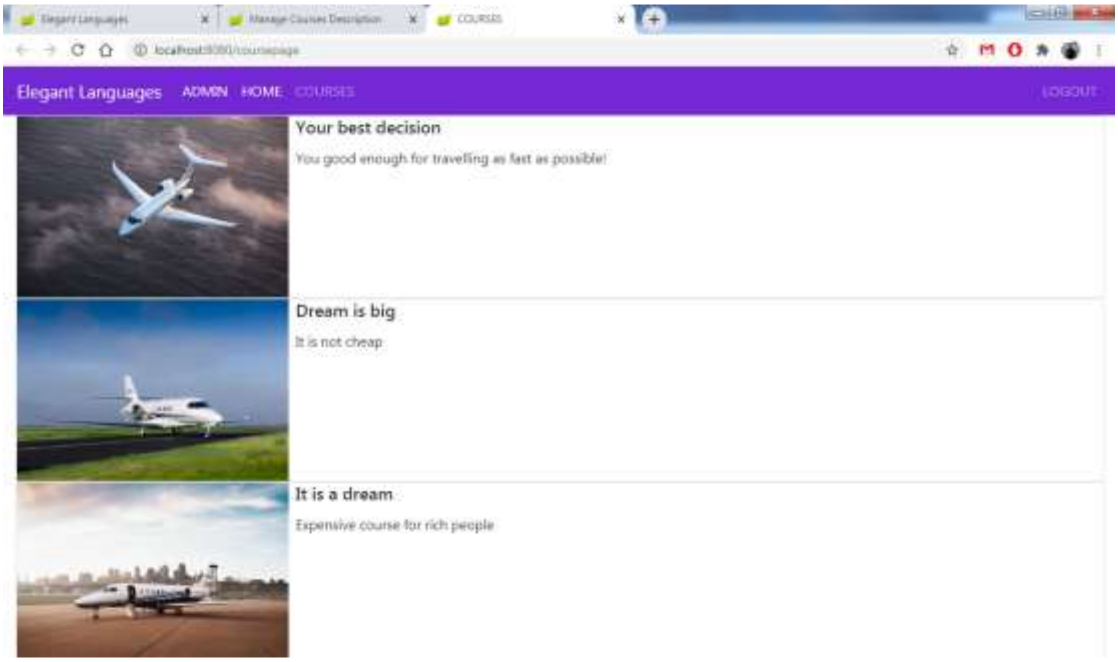

Elegant Languages
Manage Courses Description
COURSES
localhost:8080/manage_courses_descr
Elegant Languages
ADMIN
HOME
COURSES
LOGOUT
Open add courses description form
Save
Delete
Вибрати файл

List of course info

Photo	Course Title	Course Text
Longitude-7.jpg	Your best decision	You good enough for t

Fig. 3.23. Process of editing course data on Manage courses description page

Pressing Save button results in changes on courses page (fig. 3.24).



Your best decision

You good enough for travelling as fast as possible!

Dream is big

It is not cheap

It is a dream

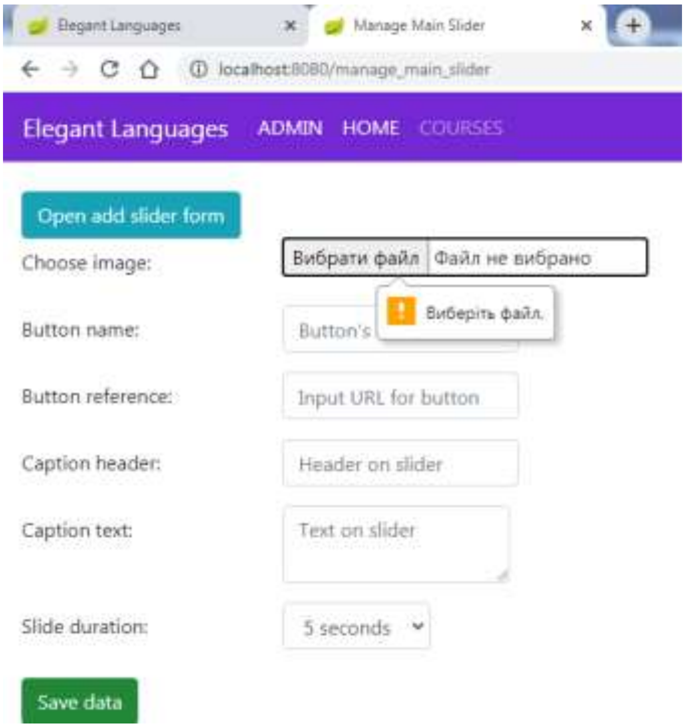
Expensive course for rich people

Fig. 3.24. Top course data changed after edition performed

3.2. Web Application Reliability

In any cases, web application should be ready to handle users’ inappropriate inputs. On main slider settings page user can try to store inappropriate data in slider form. So that there are several mechanisms are used to prevent such actions.

Firstly, slider cannot be stored without image. Pressing Save button without image chosen cause warning popup window appearing (fig. 3.25).



Elegant Languages
Manage Main Slider
localhost:8080/manage_main_slider
Elegant Languages ADMIN HOME COURSES
Open add slider form

Choose image: Вибрати файл Файл не вибрано

Виберіть файл.

Button name: Button's

Button reference: Input URL for button

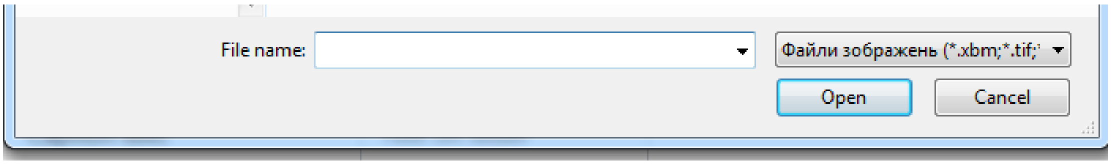
Caption header: Header on slider

Caption text: Text on slider

Slide duration: 5 seconds

Fig. 3.25. Warning message after attempt to press Save data button

With button to choose, file user potentially can choose not image file that is inappropriate. So that there are some mechanisms to prevent it is implemented. After pressing choose file, by default windows explorer shows only image files (fig. 3.26).

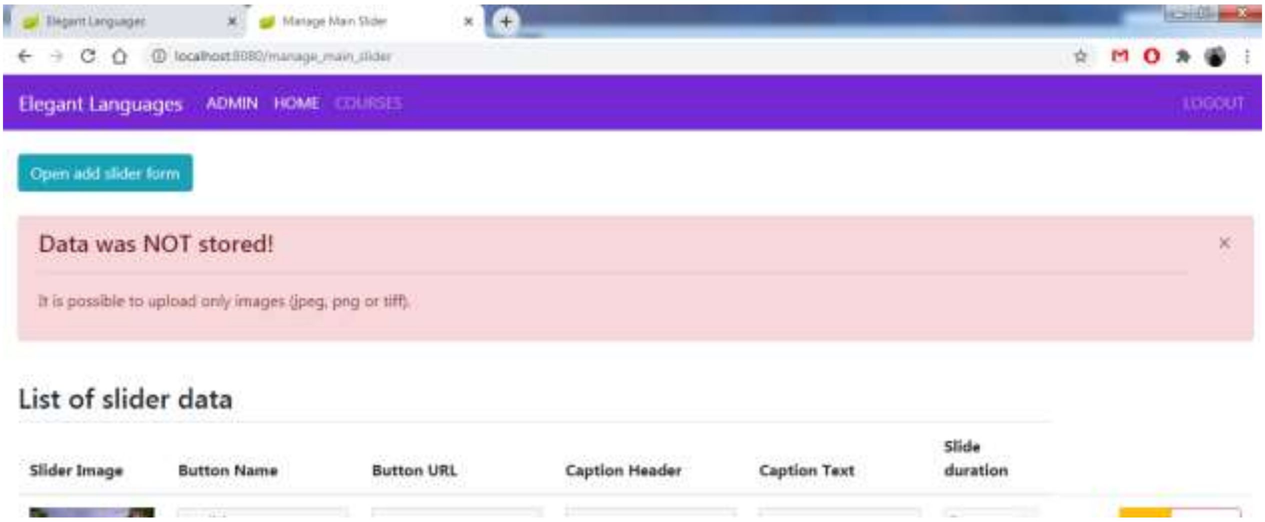


File name:

Файли зображень (*.xbm;*.tif;*

Fig. 3.27. By default, only image files can be shown for choosing

Despite of this, user can anyway change types of file to show and choose another type. In this case another mechanism is implemented. When file uploads to the backend part of application, it performs check if receiving file image or not. Depending on the result of checking, it decides to store data with returning message of successful storing or not to save data with unsuccessful message returned (fig. 3.27, fig. 3.29).



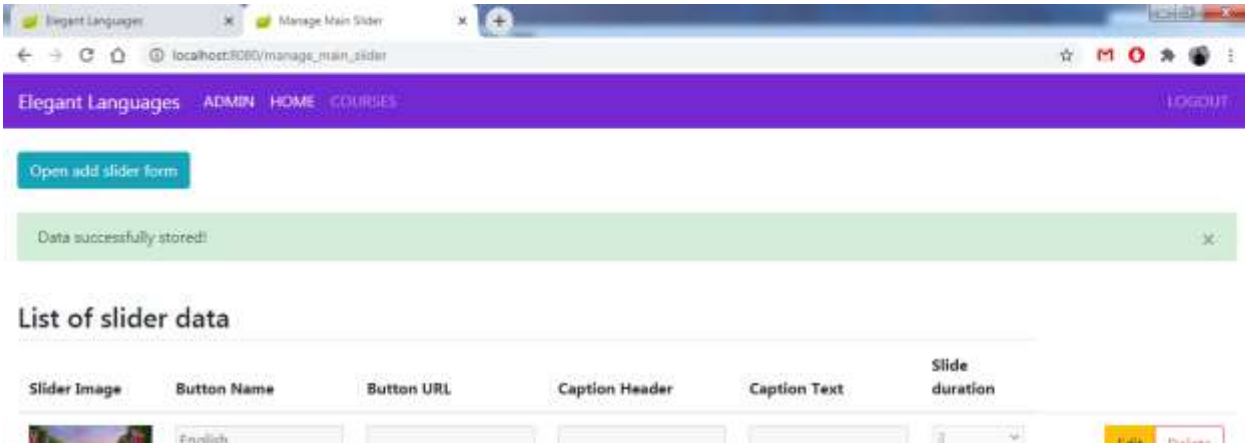
Data was NOT stored!

It is possible to upload only images (jpeg, png or tiff).

List of slider data

Slider Image	Button Name	Button URL	Caption Header	Caption Text	Slide duration

Fig. 3.27. Error message returned due to inappropriate file storing attempt



Data successfully stored!

List of slider data

Slider Image	Button Name	Button URL	Caption Header	Caption Text	Slide duration

Fig. 3.28. Message returned about successful data storing

On pages for cards and course description administration, the same mechanisms are implemented.

Conclusions of the Third Part

Created web application provides possibilities to display necessary information for school of language courses. Main page contains from navigation bar, slider and cards – special blocks with teachers' short biography. In addition, separate page with list of available courses and its descriptions available.

Nothing of all above - slides, cards and list of courses are "hardcoded". Instead, there are possibility for administrator to rule these data exist. Admin user can upload new images for slider; provide additional data on uploaded images such as pin button with link or title and main text. Admin user can delete existing slider or edit it. While edition it is possible to update all or only necessary data – image, button name, title text or description text. Only image is always required in all content. Other data is not obvious

Cards administration has the same logic. However, there are other data. Cards can contain image, title (for teachers' names), description text (for their biography) and possibility to add button with link. It is possible add or delete cards. When card was added, cards placement rearranged automatically to better suit on page. It is possible to edit all or only some part of data.

The admin user also can administrate content of courses page. However, it contains less data – only image, title and description.

Web application also provides some verification mechanisms to prevent uploading inappropriate data. Firstly, it prevent storing data without file chosen. By default, window explorer shows only image files. However, user can manually choose another type of file. On the backend, web application checks received file and decide to store data or not with status message thrown.

To sum up, web application process most data on backend. Only save/edit/delete button layout is managed on frontend.

CONCLUSIONS

During this work web application was developed. To achieve this, a big research was performed to study variety of technologies that are necessary for such kind of work.

Developing of web application demands usage of various technologies, tools and solid theoretical knowledge. Firstly, web application can not be designed without specialized tools such as integrated development environment and database management system. Then, it was necessary to understand what patterns and technologies to implement. During research, model view controller pattern was chosen for realization. There were multiple programming languages used for different parts of application. Backend development was performed using Java language. In addition, a bit JavaScript was used for some page activity. To implement dynamic page layout, Thymeleaf template engine was used. It runs on backend and bases on Java.

Webpages are user interfaces of web application. For pages creation HTML with Boostrap were used. Bootstrap is framework for simplifying developing. Layout was designed with mobility in mind. It stands for page’s dynamic suiting for different screen sizes that user uses for visiting it.

Main features of web application were fully implemented. It provides possibilities for user to manipulate main slider on home page. Admin user can add slider data, edit it and remove. Slider data can consist of different set of data – image, button or title and description text. In addition, admin user can rule special cards for teachers’ descriptions. It is possible to add, edit and remove card data. That includes images, names, descriptions and buttons with link. Another feature is placed on another page. List of courses presented separately. However, admin user is able to add, edit or remove its content.

To sum up, web application verifies files that user trying to upload for slider, card or course list. It automatically checks if uploaded file an image or not. If not, it denies storing.

REFERENCE LIST

1. Shevchenko R. Рейтинг мов програмування 2020: JavaScript випередив Java, а Dart увійшов у першу лігу (02/2020) [Internet Resources] / Website: Dou; Access mode: https://dou.ua/lenta/articles/language-rating-jan-2020/?from=doufp free.

2. JavaScript – Overview [Internet Resources] / Website: Tutorialspoint; Access mode: https://www.tutorialspoint.com/javascript/javascript_overview.htm free.

3. The Java Tutorials [Internet Resources] / Website: Oracle Java Documentation; Access mode: https://docs.oracle.com/javase/tutorial/getStarted/intro/definition.html free.

4. Everything You Need To Know About C# (05/2019) [Internet Resources] / Website: Pluralsight; Access mode: https://www.pluralsight.com/blog/software-development/everything-you-need-to-know-about-c- free.

5. Зарплати українських розробників — літо 2020 (07/2020) [Internet Resources] / Website: Dou; Access mode: https://dou.ua/lenta/articles/salary-report-devs-june-2020/?from=salaries free.

6. ReactJS - Quick Guide [Internet Resources] / Website: Tutorialspoint; Access mode: https://www.tutorialspoint.com/reactjs/reactjs_quick_guide.htm free.

7. Node.js - Quick Guide [Internet Resources] / Website: Tutorialspoint; Access mode: https://www.tutorialspoint.com/nodejs/nodejs_quick_guide.htm free.

8. What Is AngularJS? [Internet Resources] / Website: AngularJS; Access mode: https://docs.angularjs.org/guide/introduction free.

9. .NET Core - Quick Guide [Internet Resources] / Website: Tutorialspoint; Access mode: https://www.tutorialspoint.com/dotnet_core/dotnet_core_quick_guide.htm free.

10. Spring Framework - Overview [Internet Resources] / Website: Tutorialspoint; Access mode: https://www.tutorialspoint.com/spring/spring_overview.htm free.

11. The 7 Most Important Software Design Patterns [Internet Resources] / Website: Educative; Access mode: https://medium.com/educative/the-7-most-important-software-design-patterns-d60e546afb0e free.

12. Singleton [Internet Resources] / Website: Refactoring Guru; Access mode: https://refactoring.guru/design-patterns/singleton free.

13. Strategy [Internet Resources] / Website: Refactoring Guru; Access mode: https://refactoring.guru/design-patterns/strategy free.

14. Observer [Internet Resources] / Website: Refactoring Guru; Access mode: https://refactoring.guru/design-patterns/observer free.

15. Design Patterns - MVC Pattern [Internet Resources] / Website: Tutorialspoint; Access mode: https://www.tutorialspoint.com/design_pattern/mvc_pattern.htm free.

16. MVC Design Pattern (08/2018) [Internet Resources] / Website: GeeksforGeeks; Access mode: https://www.geeksforgeeks.org/mvc-design-pattern/ free.

17. Rouse M. Database (DB) [Internet Resources] / Website: TechTarget; Access mode: https://searchsqlserver.techtarget.com/definition/database free.

18. What is a Relational Database Management System? [Internet Resources] / Website: Codecademy; Access mode: https://www.codecademy.com/articles/what-is-rdbms-sql free.

19. Thakur A. 10 Best CSS Frameworks for Front-End Developers (05/2020) [Internet Resources] / Website: Geekflare; Access mode: https://geekflare.com/best-css-frameworks/ free.

20. Chand M. Most Popular Databases In The World (07/2019) [Internet Resources] / Website: C# Corner; Access mode: https://www.c-sharpcorner.com/article/what-is-the-most-popular-database-in-the-world/ free.

21. Thymeleaf FAQ [Internet Resources] / Website: Thymeleaf; Access mode: https://www.thymeleaf.org/faq.html free.

22. IntelliJ IDEA [Internet Resources] / Website: JetBrains; Access mode: https://www.jetbrains.com/idea/download/#section=windows free.

23. Lsnrctl [Internet Resources] / Website: Oracle FAQ’s; Access mode: https://www.orafaq.com/wiki/Lsnrctl free.

24. Listener [Internet Resources] / Website: Oracle FAQ’s; Access mode: https://www.orafaq.com/wiki/Listener free.

25. Guide to naming conventions on groupId, artifactId, and version (12/2020) [Internet Resources] / Website: Apache Maven Project; Access mode: https://maven.apache.org/guides/mini/guide-naming-conventions.html free.

26. Java Language Specification Chapter 7 Packages [Internet Resources] / Website: Oracle Help Center; Access mode: https://docs.oracle.com/javase/specs/jls/se6/html/packages.html free.

27. Core J2EE Patterns - Data Access Object [Internet Resources] / Website: Oracle; Access mode: https://www.oracle.com/java/technologies/dataaccessobject.html free.

28. Spring Boot JpaRepository tutorial (07/2020) [Internet Resources] / Website: ZetCode; Access mode: http://zetcode.com/springboot/jparepository/ free.