MUNYANEZA KENNY ROGER

ID: 23043

DATE: 15/05/2023

Roads and Utilities Inspection Management Application Documentation

Content

1. Overview

This project aims at facilitating the task of inspection done on all construction and installations of roads and utilities projects. This application shall be used by the office of the city of Kigali. It will be used by inspectors during the inspection of all construction projects that take place in Kigali city. Also, project managers in charge of specific projects will capture all the details about the project at initiation phase.

Lastly, the operations manager shall use this system to manage inspectors and projects managers that are signed up to the system as a way of ensuring that all projects are being developed with adequate follow up.

2. Requirements of the application

Functional requirements:

1.User authentication: The system should have a secure login mechanism to authenticate inspectors, project managers, and operations managers.

2.Project initiation: Project managers should be able to enter all the necessary details about a construction project during the initiation phase, including project name, location, description, timeline, and other relevant information.

3.Inspection reporting: Inspectors should be able to record and submit inspection reports, including details of observations, findings, non-compliance issues, and recommendations for corrective actions.

4. Project monitoring: Operations managers should have the ability to monitor the progress of construction projects, view inspection reports, and track compliance with standards and regulations.

5. Inspector and project manager management: The system should provide functionality for operations managers to manage the list of inspectors and project managers, including adding, removing, and assigning them to specific projects.

Non-functional requirements:

1. Security: The system should have robust security measures in place to protect user data, prevent unauthorized access, and ensure data privacy.
2. Usability: The user interface should be intuitive and user-friendly, allowing inspectors, project managers, and operations managers to easily navigate and perform their tasks.
3. Reliability: The system should be reliable and available for use at all times, with minimal disruptions or downtime.
4. Scalability: The system should be scalable to accommodate the growing number of construction projects, inspectors, and project managers in the future.
5. Data backup and recovery: The system should regularly backup data to prevent data loss and have mechanisms in place for data recovery in case of any failures or disasters.
6. Validation: The system user-interfaces should validate data that is being input to prevent capturing wrong, irrelevant or inaccurate information.

3. Project Plan

Scope

The scope of the project is limited to the inspection of roads and utilities construction projects in the city of Kigali. I will be capturing the address of projects sites that are relevant to the addresses in Kigali, namely the addresses that start with KK for Kicukiro district projects, KG for Gasabo district projects, or KN for Nyarugenge district projects. Also, the application's scope is limited to the capturing the construction projects sites and updating the records in the application gradually as the construction works go by and the project progresses.

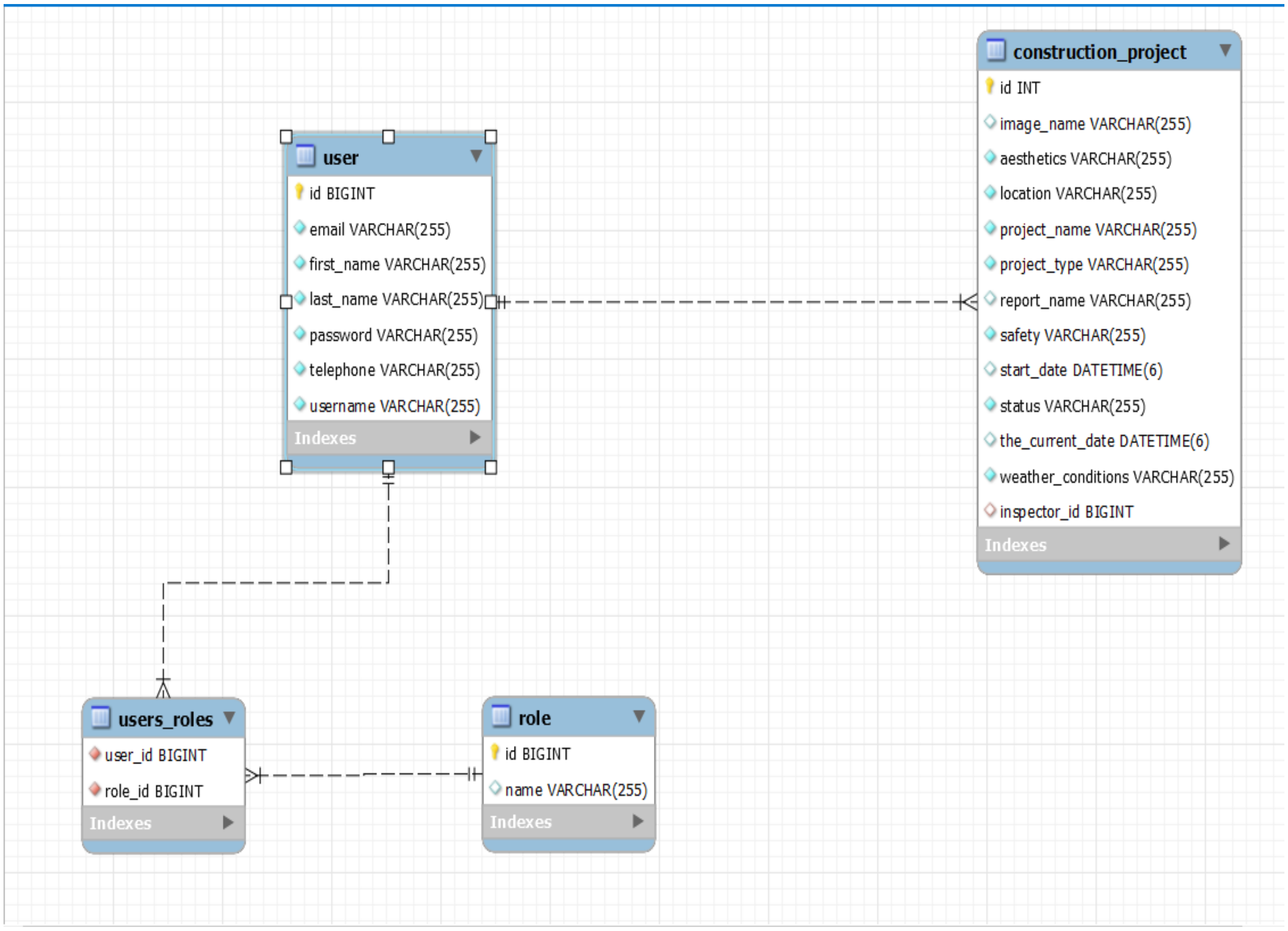
Timeline

This application was developed in a period of two weeks, whereby the requirements definition took one day, designing took two days, design implementation and coding took nine days, testing took one day and deployment took one day.

Resources

I used web technology resources to develop this application. The resources are divided into programming languages like Java, SQL, JavaScript, and CSS; markup languages like HTML; frameworks like Spring MVC, Spring Boot, Spring Data JPA; Servers like Tom Cat and MySQL server; MySQL Workbench client application; and lastly a development environment called Intellij IDEA.

4. Database Schema



- Users are mapped to their role by ManyToMany relationship.
- Each Construction Project is mapped to the responsible inspector by ManyToOne relationship.

5. User Documentation

Access the application using the URL: https://inspection-project-final.herokuapp.com/

Login:

Possible credentials

Role: PROJECT_MANAGER

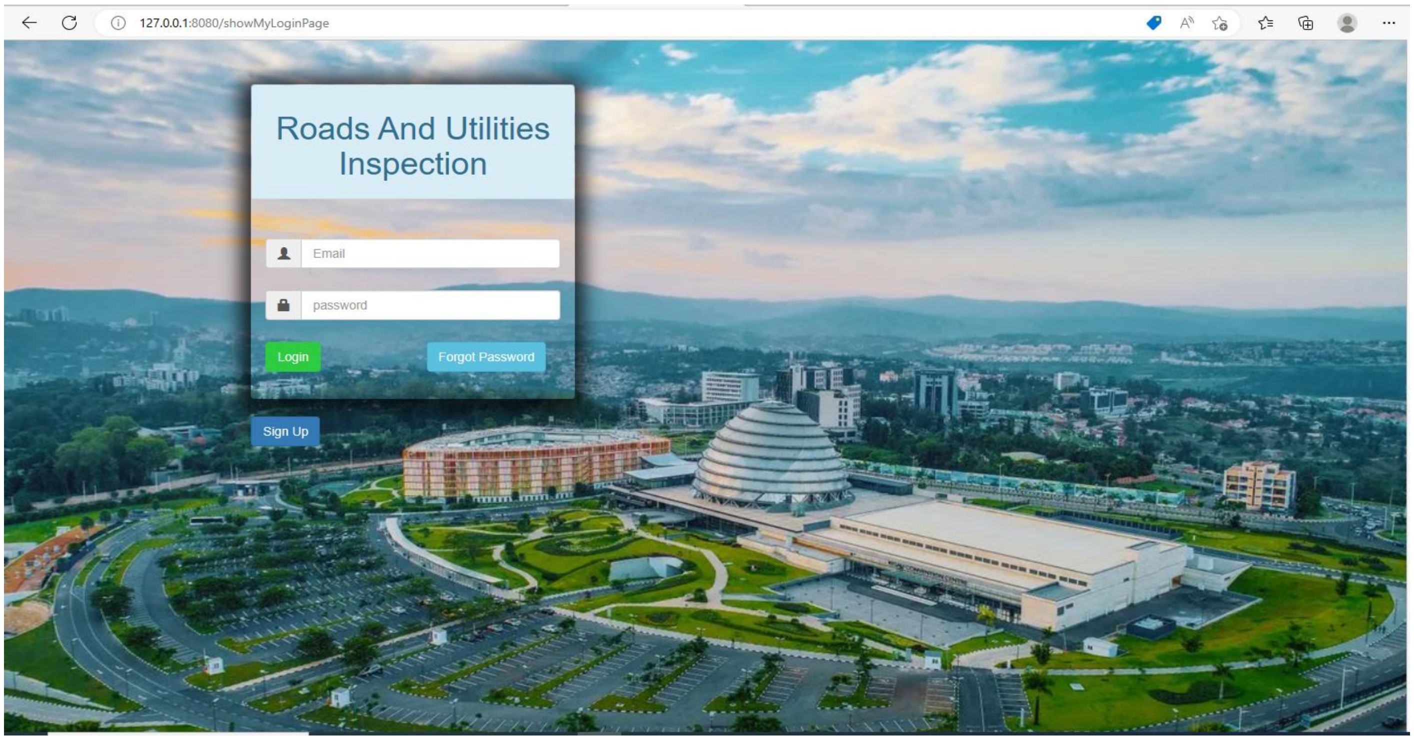
- Email: andy@gmail.com
- Password: try123
- Phone number (required to reset password): 0785048106

Role: INSPECTOR

- Email: willy@gmail.com
- Password: auca@2021
- Phone number (required to reset password): 0785048106

Role: OPERATIONS_MANAGER

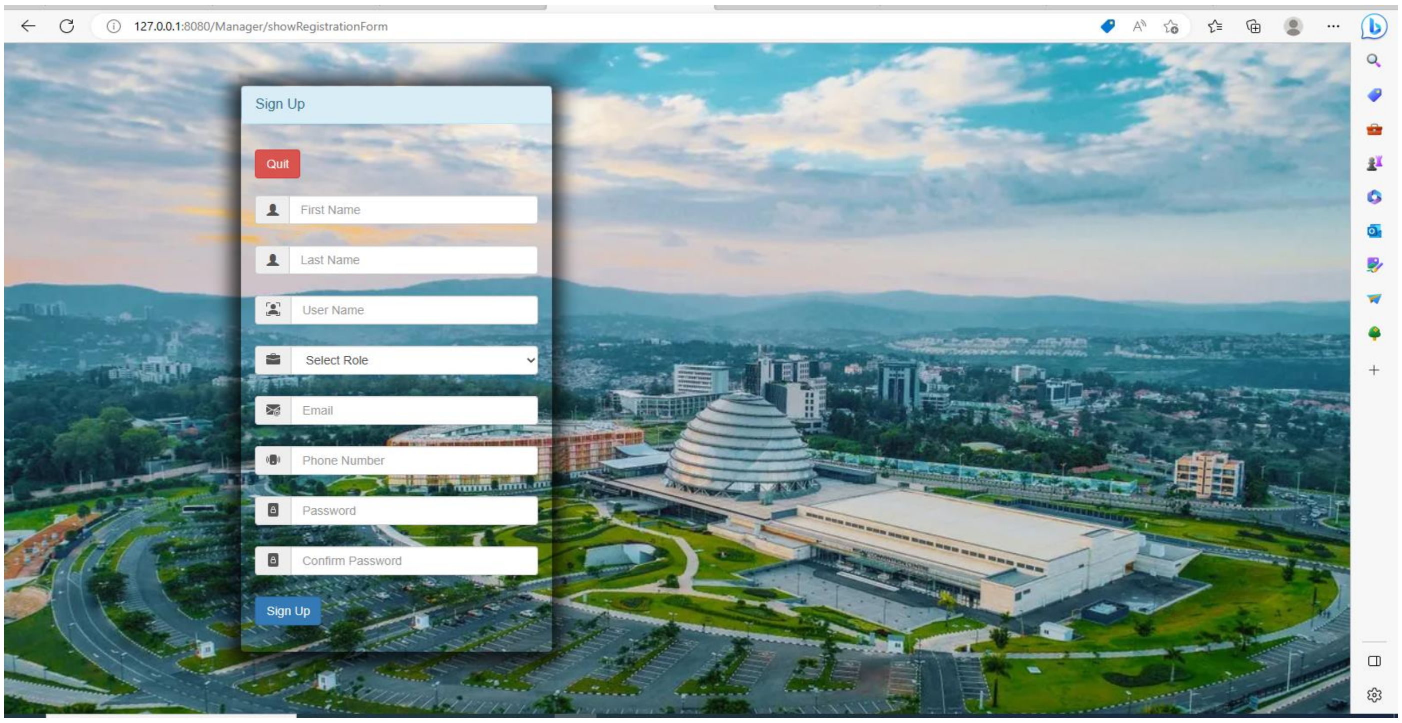
- Email: kennyrogers330@gmail.com
- Password: new123
- Phone number (required to reset password): 0781885227


127.0.0.1:8080/showMyLoginPage
Roads And Utilities Inspection
Email
password
Login
Forgot Password
Sign Up

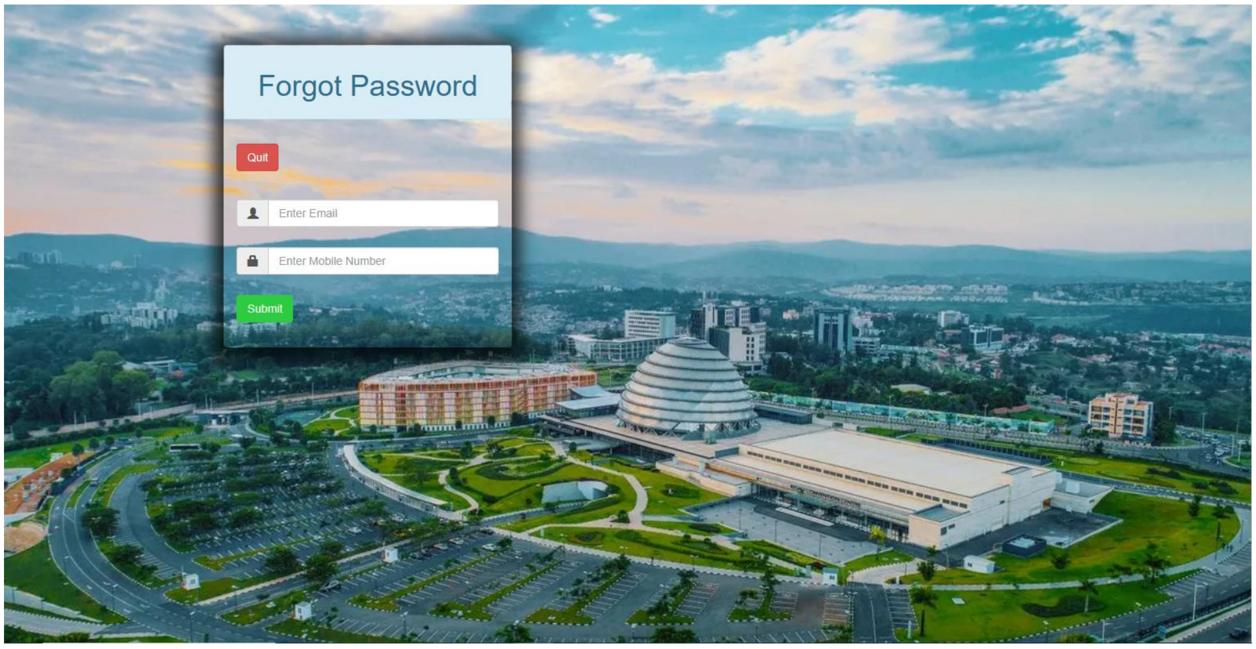
<u>Sign Up form:</u>

This form may be used by a new user to sign up with the relevant roles they will have on the system.

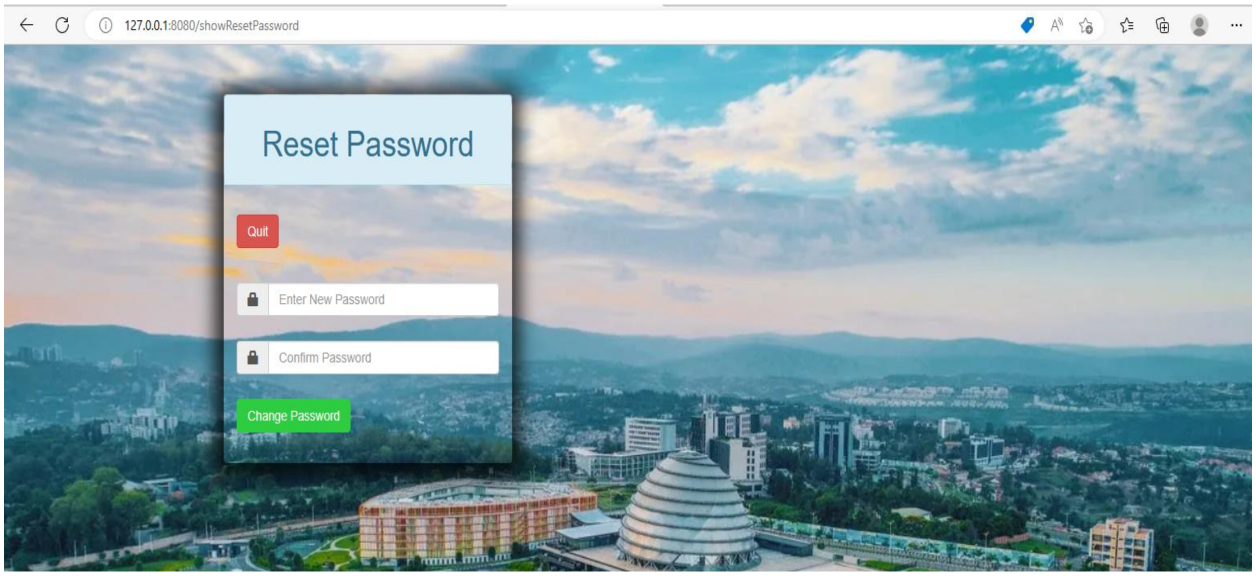
To be able to sign Up successfully, the user must meet all the form validations required.



In case the user forgot their password, they are required to provide their email and phone number that are associated with their account.



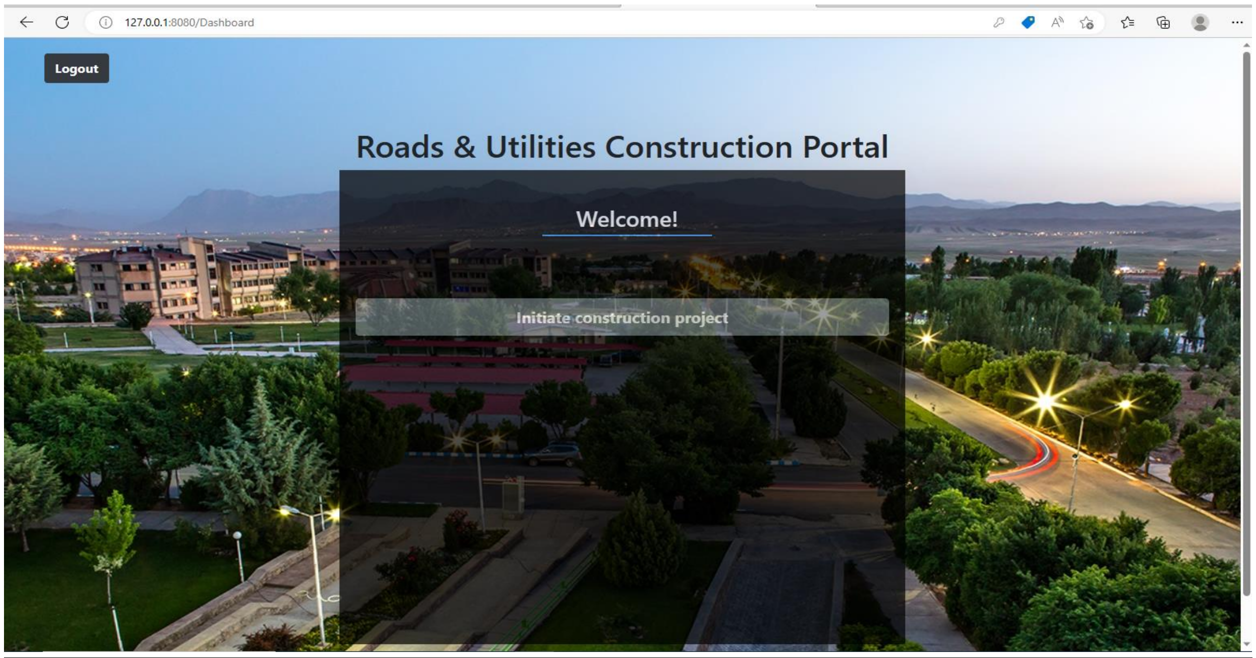
After, the user may get a form to add a new password that shall be used on the next logins.



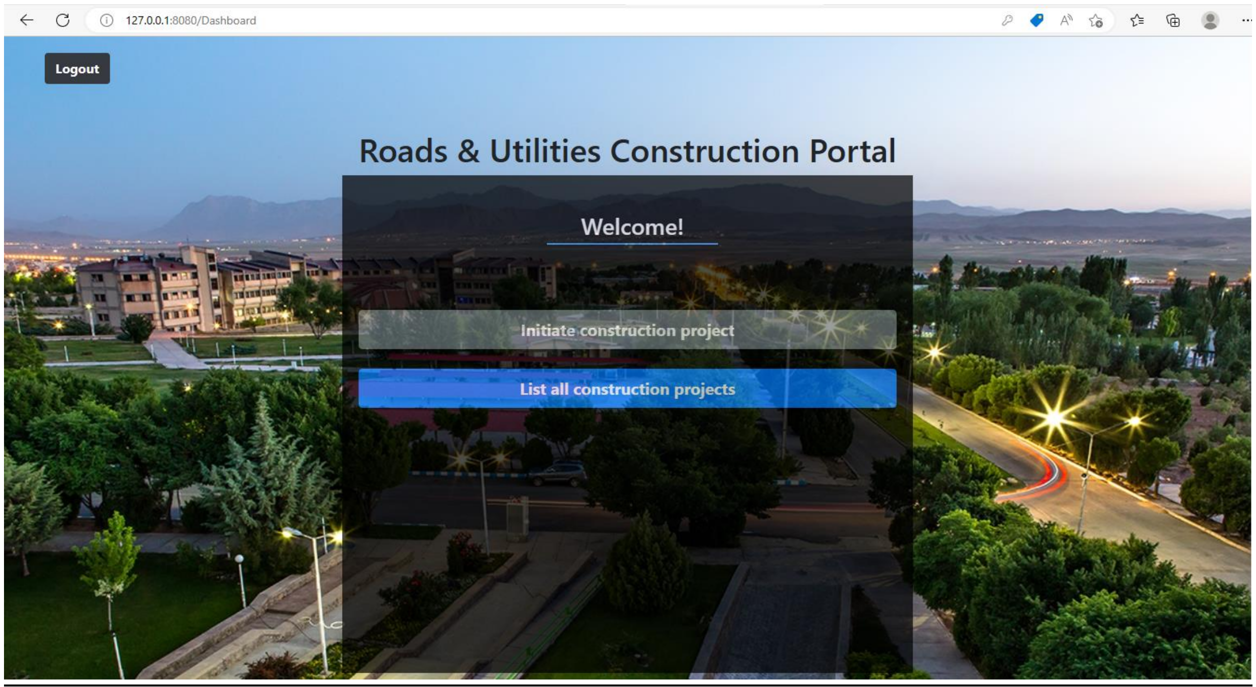
System Dashboard:

This dashboard was developed leveraging spring security to establish view based on roles feature.

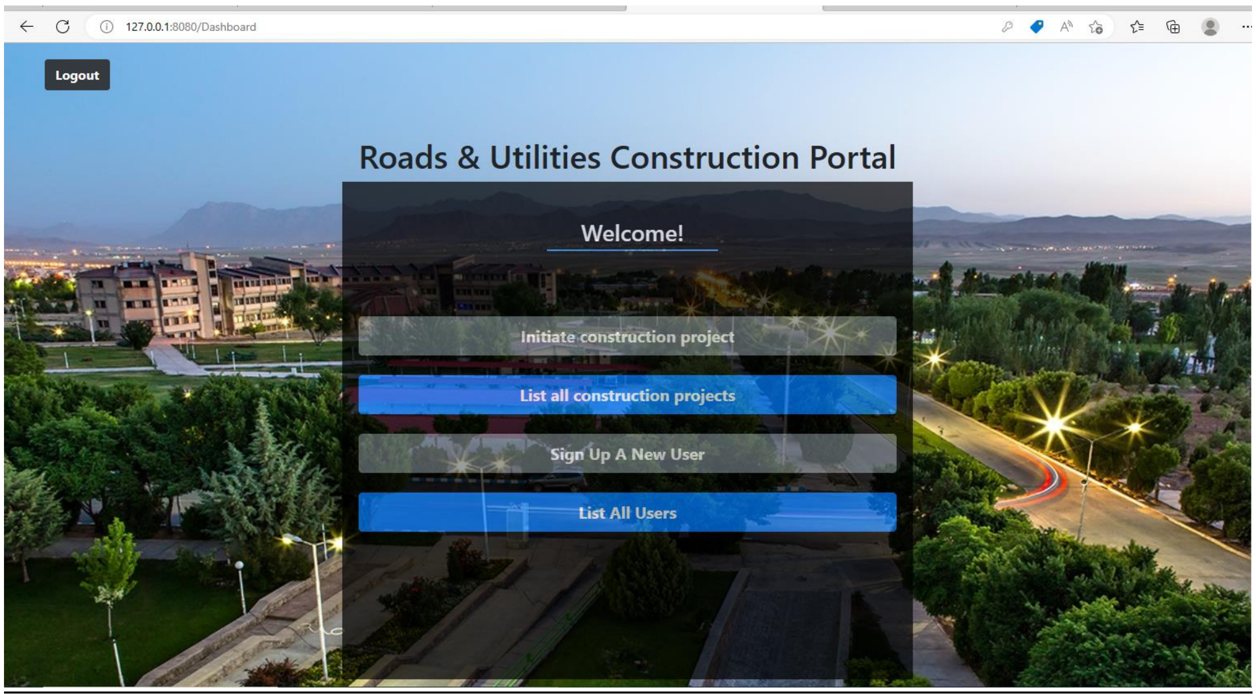
Project Manager's Dashboard


127.0.0.1:8080/Dashboard
Logout
Roads & Utilities Construction Portal
Welcome!
Initiate construction project

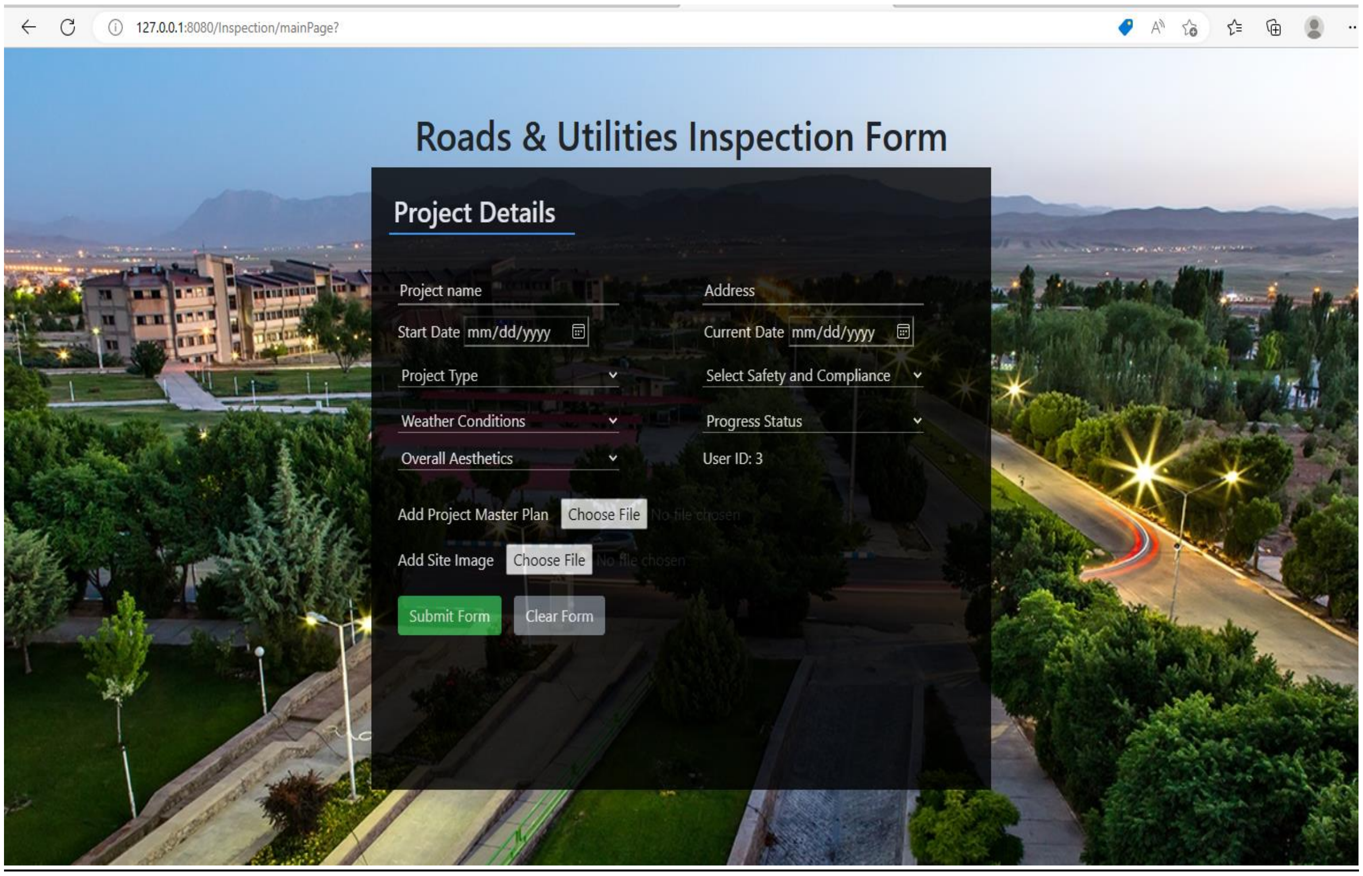
Inspector 's Dashboard


127.0.0.1:8080/Dashboard
Logout
Roads & Utilities Construction Portal
Welcome!
Initiate construction project
List all construction projects

Operations Manager 's Dashboard


127.0.0.1:8080/Dashboard
Logout
Roads & Utilities Construction Portal
Welcome!
Initiate construction project
List all construction projects
Sign Up A New User
List All Users

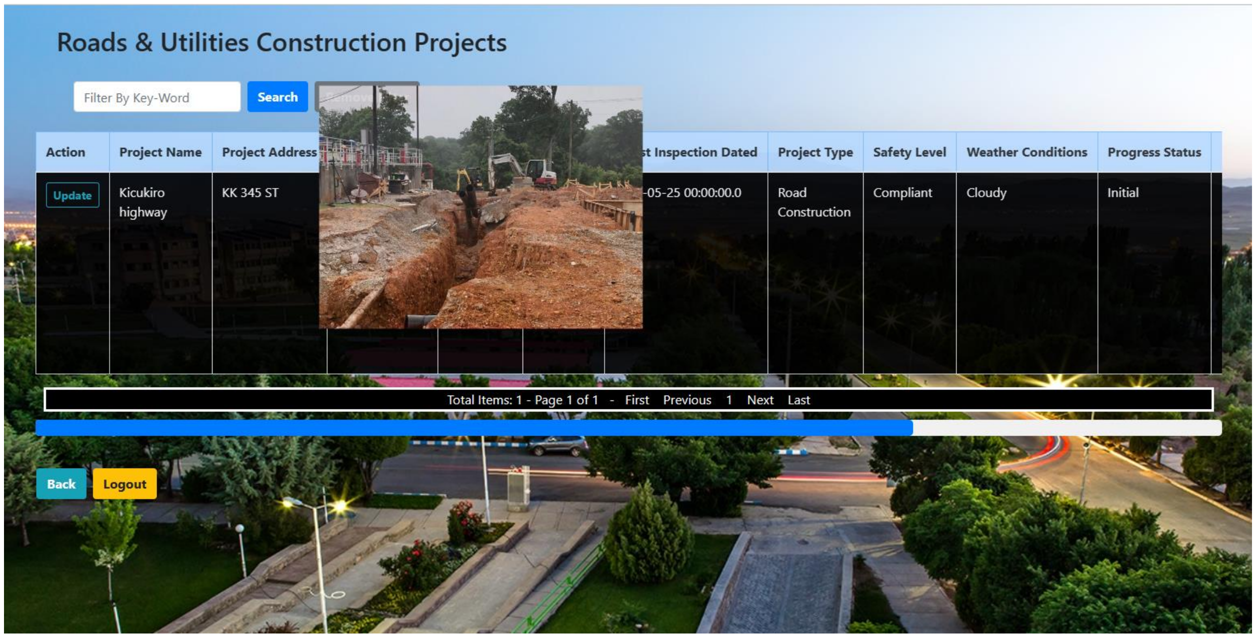
Project initiation form for Project Managers



This form will allow a user to initiate a construction project and record the initial state of the project by specifying the project name, the project start date, the project type, the image of the site at start of the project as well as the masterplan of the project. This page is prepopulated with the id of the logged in user that is capturing the new project.

Listing All construction and establishment projects

Since the application is an inspection driven application, the minimum privileges are given to the project managers that carry out the project, therefore they can only capture the project at initiation stage on the application. They are not allowed to access any further resources on the application such as listing all construction projects, updating the state of the infrastructure sites or deleting construction projects.

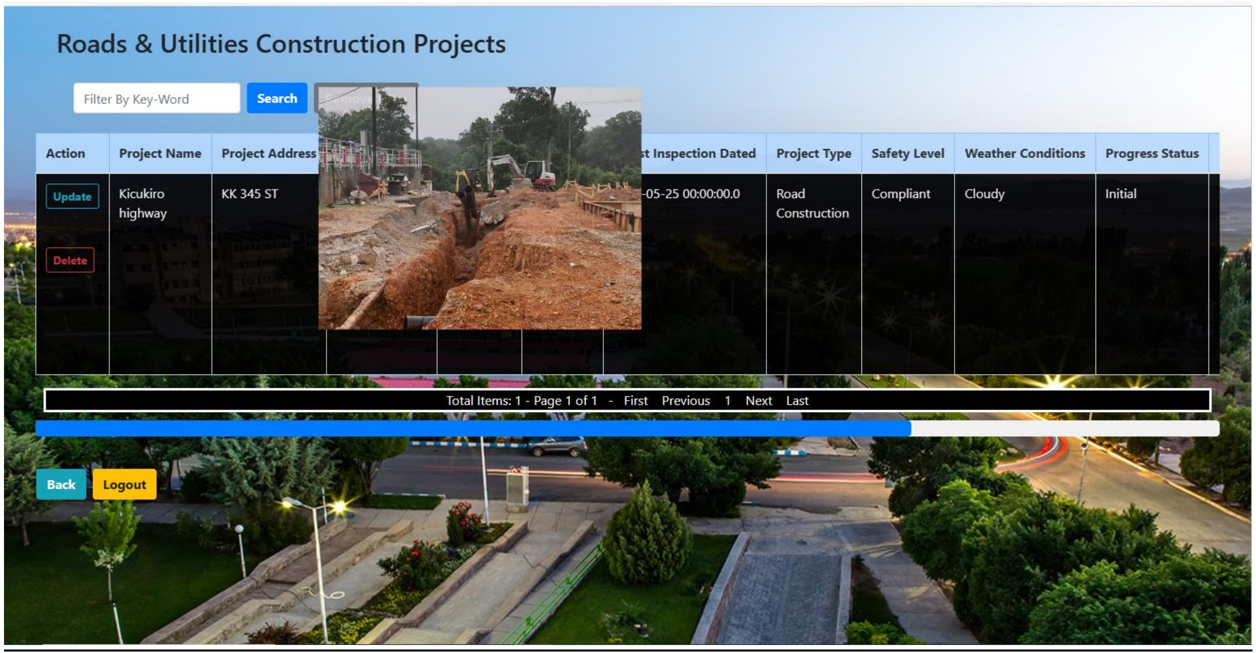

Roads & Utilities Construction Projects

Action	Project Name	Project Address	t Inspection Dated	Project Type	Safety Level	Weather Conditions	Progress Status
Update	Kicukiro highway	KK 345 ST	-05-25 00:00:00.0	Road Construction	Compliant	Cloudy	Initial

Total Items: 1 - Page 1 of 1 - First Previous 1 Next Last

When you list all construction projects, you will get a list of all projects that were captured on the system. The list provided has the pagination feature which lists two projects on each page. You can navigate to other pages of the list using the next, last, previous, first hyperlinks under the list table. There's also a search and filtering feature, whereby the user may provide a keyword and the list of all matching data will be provided.

NOTE: Only users signed in as INSPECTOR or OPERATIONS_MANAGER can see the **update** button. Also, the **delete** button is not visible to inspectors because only the operations manager are allowed to delete projects on the system. The image below shows the list of projects whereby delete is possible because the signed user has role of OPERATIONS_MANAGER.



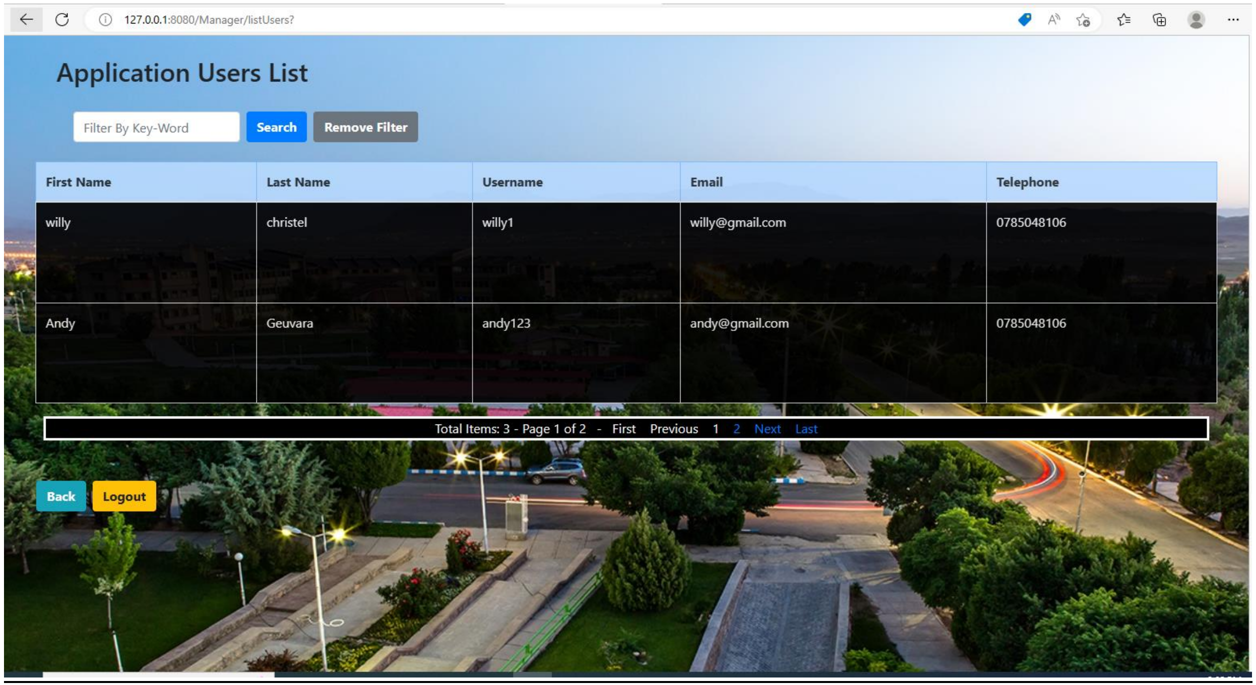
Roads & Utilities Construction Projects

Action	Project Name	Project Address	t Inspection Dated	Project Type	Safety Level	Weather Conditions	Progress Status
Update Delete	Kicukiro highway	KK 345 ST	-05-25 00:00:00.0	Road Construction	Compliant	Cloudy	Initial

Total Items: 1 - Page 1 of 1 - First Previous 1 Next Last

List All users:

Only the operations manager is given a button on the dashboard that allows him/her to list all users that are signed up on the application to view all their credentials.



127.0.0.1:8080/Manager/listUsers?

Application Users List

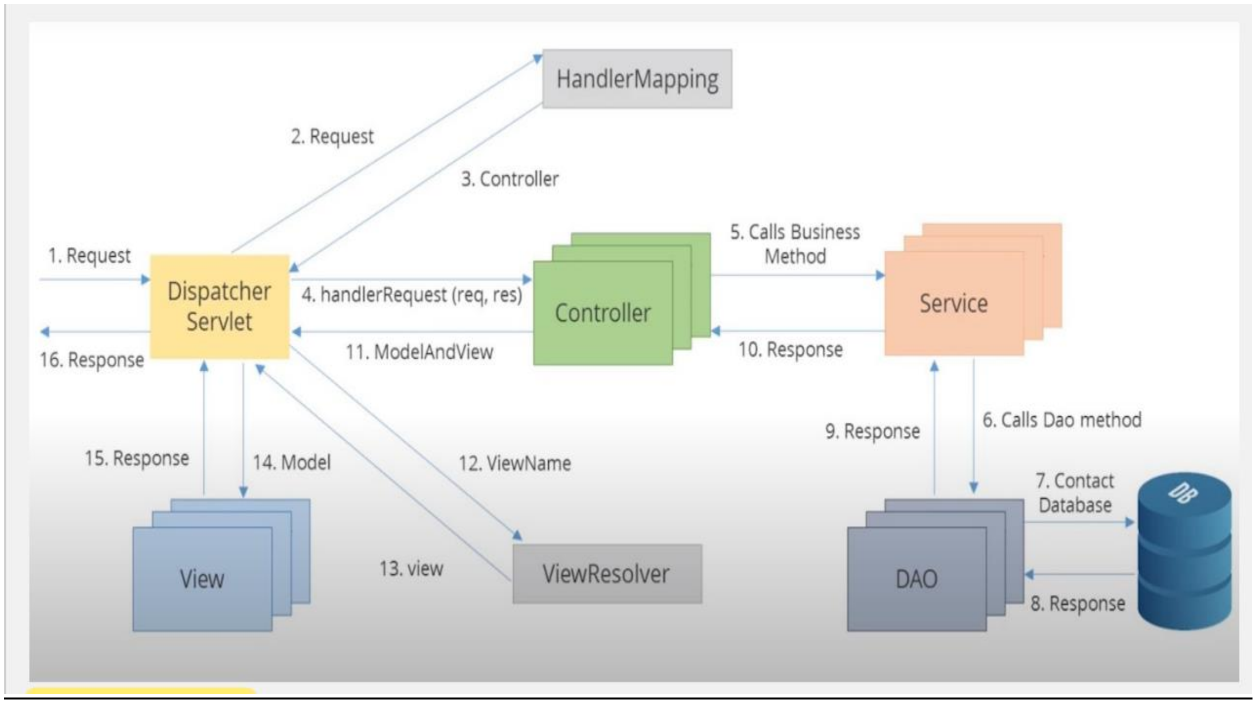
First Name	Last Name	Username	Email	Telephone
willy	christel	willy1	willy@gmail.com	0785048106
Andy	Geuvara	andy123	andy@gmail.com	0785048106

Total Items: 3 - Page 1 of 2 - First Previous 1 2 Next Last

6. Technical Documentation

Architecture of the application

I used spring MVC architecture that stands for spring Model, View and Controller as the architecture of the application.



To implement this architecture, I developed a springboot maven application.

- I added a dependency of "**spring-boot-starter-web**". This dependency includes the necessary libraries for creating web applications with Spring Boot, including the DispatcherServlet. The DispatcherServlet is automatically registered and configured when you include the necessary dependencies in your Maven pom.xml file and annotate your application's main class with @SpringBootApplication.

- I added a dependency of "**spring-boot-starter-thymeleaf**". The **Handler-Mapping** is responsible for mapping incoming requests to the appropriate controller methods based on defined mappings and rules. It helps determine which controller should handle a specific request URL. When you include the spring-boot-starter-web and spring-boot-starter-thymeleaf dependencies in your Maven pom.xml file, Spring Boot automatically configures the Handler-Mapping as part of its auto-configuration.

- I have a three **Controller** classes namely, "**LoginController**", "**ManagerController**", and "**MainController**". The Manager controller is used to handle requests that are related to signing up new users to the system and setting their roles, the Manager controller is used to handle requests related to logging in, and password resetting. The Main controller is used to handle requests related to CRUD operations that are to be performed on the main domain model which is the construction projects.

- I have the **service layers** that is an intermediate layer between the controller layer and the data access layer (typically represented by repositories or DAOs). Its main purpose is to encapsulate the business logic of the application and provide a separation of concerns. In my application, I used the userService layer to handle the business logic related to users of the application. Also, I have a projectService layer that handles the business logic related to construction projects persisted on the application.

- I have used the Spring Data JPA framework mainly the interface called **JpaRepository** by adding the dependency "**spring-boot-starter-data-jpa**". JpaRepository is one of the central interfaces provided by Spring Data JPA. It extends the standard CrudRepository interface and provides additional methods for common database operations such as querying, saving, updating, and deleting entities. JpaRepository also supports pagination, sorting, and other convenience features.

- The **view resolver** in Spring Boot resolves the logical view names returned by the controller methods to the actual view templates that will be rendered and returned to the client. When you include the **spring-boot-starter-thymeleaf** dependency in Maven pom.xml file, Spring Boot automatically configures the view resolver as part of its auto-configuration.

- **The log-back framework**. Logback is a standalone logging framework for Java applications that provides a flexible and reliable logging solution with many useful features.

- **Spring-boot-starter-validation** is a starter module in Spring Boot that provides support for validation using the Java Bean Validation API. It simplifies the configuration and usage of validation in your Spring Boot application by automatically setting up the necessary components.

- **Spring Cache:** Spring Cache is a caching abstraction provided by the Spring Framework. It allows you to add caching annotations, such as @Cacheable, @CachePut, and @CacheEvict, to your methods.

- **Spring Security:** Spring Security is a comprehensive security framework for Java applications. It offers a wide range of features for securing web applications, including authentication, authorization, and various security mechanisms.

- **MySQL-connector-java:** mysql-connector-java (MySQL Connector/J) is the official JDBC driver for connecting Java applications to MySQL databases.