Names: Mpunga Simon Pierre

ID: 22984

Date: 16th December 2023

WEBTECH FINAL PROJECT

TOPIC: ONLINE GROCERY STORE MANAGEMENT SYSTEM

1. **Project Requirement**
2. **Purpose of the project**

The purpose of online grocery store management system is to provide a digital platform for customers to conveniently browse, select, and purchase groceries over the internet. This system aims to streamline and enhance the overall grocery shopping experience by allowing users to access a diverse range of products, compare prices, receive personalized recommendations, and place orders online. It facilitates efficient inventory management for the store, enabling real-time tracking of product availability, managing stock levels, and automating order processing. Additionally, the system typically incorporates secure payment gateways, order tracking, and delivery scheduling features, offering customers the flexibility of doorstep delivery or store pickup. Overall, an online grocery store management system seeks to modernize and simplify the grocery retail process, providing a seamless and time-saving solution for both consumers and store operators

1. **Expected outcomes of the project**

The expected outcomes of the online grocery store management system project include providing customers with a convenient and personalized shopping experience, ensuring efficient inventory management and order processing, implementing secure and seamless online transactions, optimizing delivery services, enhancing operational efficiency, fostering scalability and adaptability to changing business needs, and ultimately driving positive business impact through increased sales, customer satisfaction, and data-driven decision-making. The project aims to modernize grocery retail, making it more accessible, efficient, and tailored to individual preferences while positioning the store competitively in the market

1. **Key Functional Requirement of the project**

* **User Registration and Profiles:** Users sign up as buyers, managers, or employers, creating profiles with personal info and contact details. Buyers add preferences and order history. Managers specify roles in overseeing operations. Employers include professional details. This tailored approach enhances engagement and functionality.
* **Grocery Stock Listings:** Employer can add Grocery stock and all users can see the list of all available Grocery stocks.
* **Grocery Stock Search and Filtering:** this platform allow a buyer to search for a certain grocery stock he/she want to order. And also, he/she can do filter.

* **Order Submission:** a buyer can order the groceries he/she want, where an employer receive an email that he/she has made an order.
* **Dashboard:** The platform typically includes dashboard that allows administrators to manage user accounts, grocery stock listings, site settings, and monitor system activities.
* **Login Page:** the platform allows the user to access the login page which redirect you to the dashboard according to user roles.
* **Forgot and Reset Password:** the platform allow user to reset his/her password when he/she forgot it by confirming his/her email and phone number then he/she can set the new password.
* **Pagination:** the platform will allow the user to choose the page he/she want to load and he/she can navigate the page by using next, previous, last and first page.
* **Error- Handling**: the platform will allow user to see the error messages if the platform catch an error and exception.
* **File upload and Download:** the platform will allow the user to download and upload his/her documents.
* **Validation:** the platform will allow the user to catch error messages when he/she fills in with incorrect credentials or data.

1. **Key Non-Functional Requirement of the project**

* **Reliability:** buyers can access their resume 98% of the time without failure.
* **Availability**: System should be highly available and accessible for users anytime the buyer needs to make an order. The system must always be available 24/7 and remain responsive despite failures which may occur.
* **Data integrity:** the system shall maintain data integrity by keeping backup of all updates to the database for every record transaction
* **Usability:** the UI has to be user-friendly and easy to use
* **Compatibility:** The system should be compatible with different operating systems and software platforms, like browsers, since a customer can access it using any type of operating system or/and browsing software. Such as Windows OS, Ubuntu OS, Android OS, iPhone OS, etc.
* **Recoverability:** The system should be able to Recover data if it’s lost, backup must exist
* **Performance:** The User Interface (web site)’s load time should not be more than 3 seconds for users
* **Robustness**: A robust system is able to handle error conditions gracefully, without failure. This includes a tolerance of invalid data, software defects, and unexpected operating conditions.

**ii. Project Plan**

1. **SCOPE**

The project's scope is to develop an online grocery store system with user-friendly registration, profile creation, and role-specific features for buyers, managers, and employers. It includes secure accounts, efficient inventory and order management, scalable functionalities, secure payments, optimized delivery, and data analytics. The goal is to enhance the grocery shopping experience and boost operational efficiency.

1. **TIMELINE**

This Grocery platform application was developed in a period of two weeks (14days), requirement gathering took me 2 day, design and prototyping took me 3 days, development and coding took me 6 days, testing the application 2 days and the deployment took me 1 day.

1. **Resources**

Resources I used are:

**Programming Languages and Frameworks:**

Backend Development: JAVA

Frontend Development: HTML/CSS, JavaScript, THYMELEAF Engine.

Database Management Systems: MySQL Work Bench, SQL

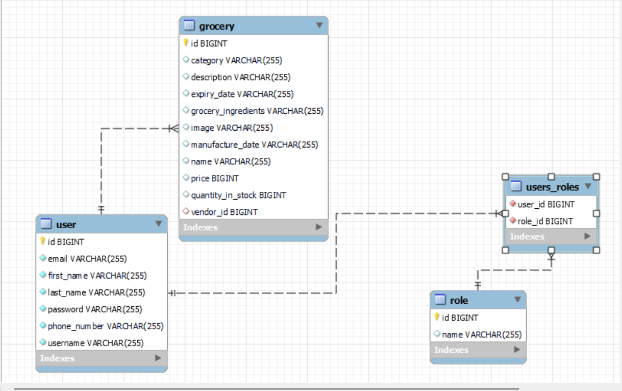
Framework: Spring MVC, Spring Boot, Spring Data JPA

**Web Development Technologies and Tools:**

Integrated development environments (IDEs) (e.g., IntelliJ, Visual Studio Code)

**Hosting platforms** (e.g., Heroku)

**Iv. Database Schema**



Users are mapped to their role by ManyToMany relationship.Each

Grocery is mapped to the responsible Employer by ManyToOne relationship.

**V . User Documentation**

**Login credentials**

**Role: ADMIN**

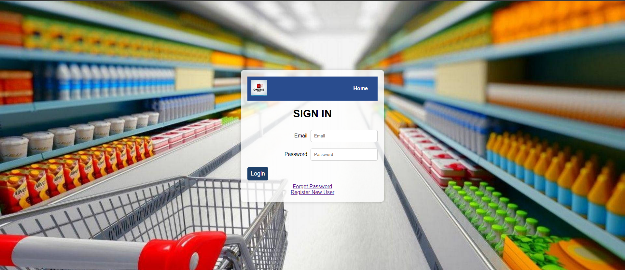
* Email: musabyimanaimmaculee02@gmail.com
* Password: fun123

To access my Grocery Board Platform use this link : https://simon-final-project.up.railway.app/

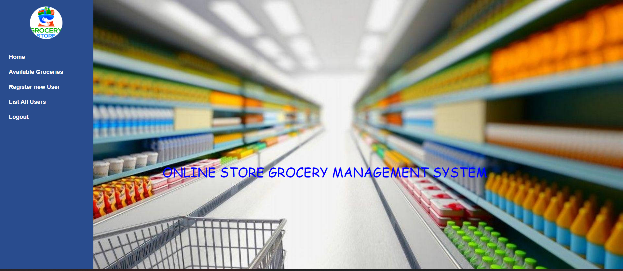
This will take you to the Login Page of the admin

* Email : musabyimanaimmaculee02@gmail.com

Password: fun123

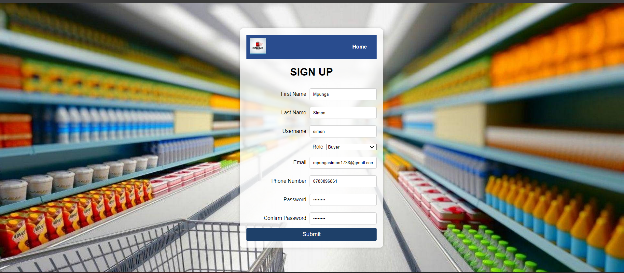


**Role Operations Manager(Admin) Dashboard**

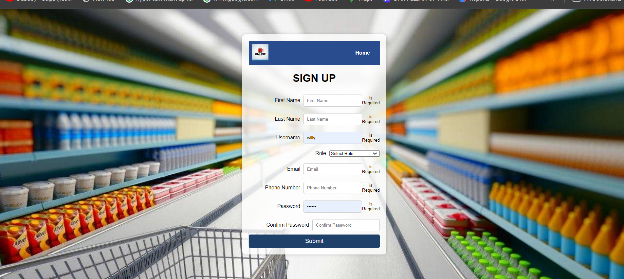
****

he/she will be able to see available jobs, add job and register new User

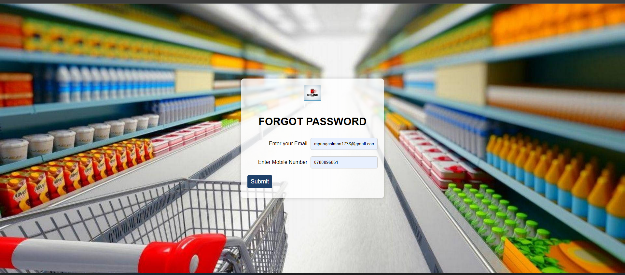
Register new user form



When you try to submit the invalid form in the register form you will receive error messages because of spring Validation. The user with the role of operations manager is the on to register new user.

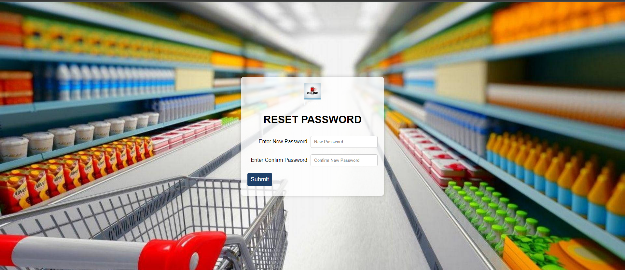


On the Login form there is an option for a user who forgot his/her password he/she will click the link of forgot password. He/she will be taken to the page for confirming if it’s his/her account by filling email and phone number

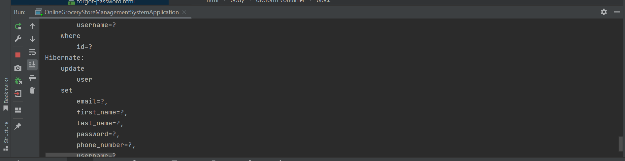


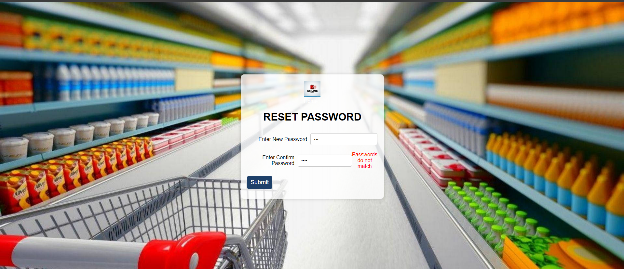
Click forgot password

If you put wrong credentials it will display invalid email and phone number ,and if the credentials are correct it will redirect you to the reset password to set new password



And if you put new password which doesn’t meet the confirm password you will get an error message



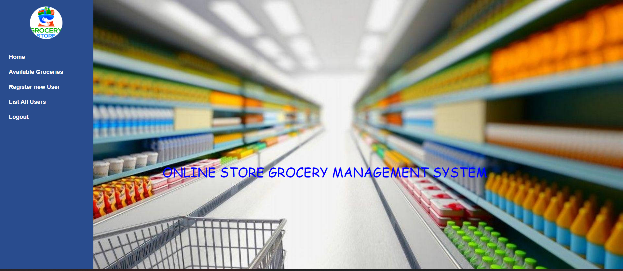


And if your password matches it will be updated in the database

**Accessing the Dashboard**

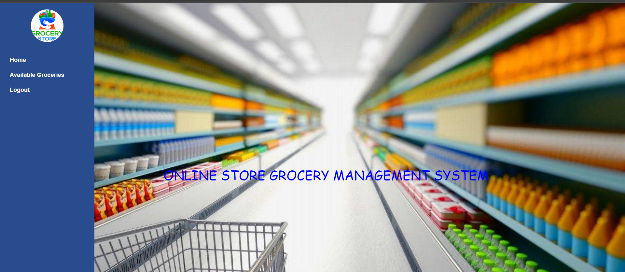
Using Spring role based security users will get different dashboard because of their roles if they try to Login

**Role Operations Manager(Admin) Dashboard**

****

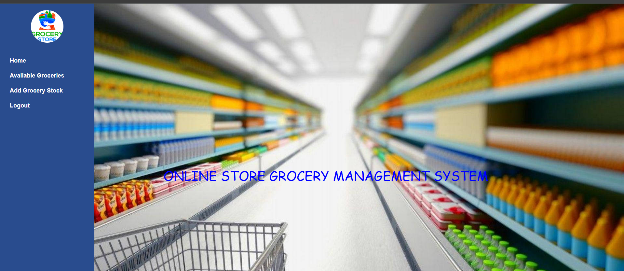
he/she will be able to see available jobs, add job and register new User

**Role Buyer Dashboard**

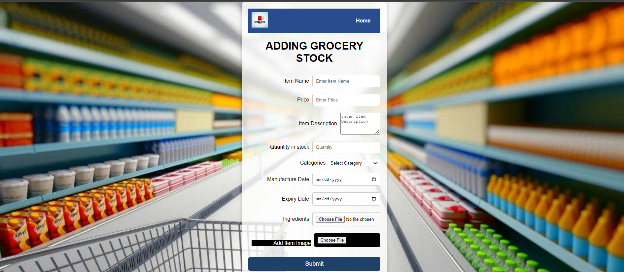
****

He/she will only have access to view available Grocery Stock only

**Role Vendor Dashboard**

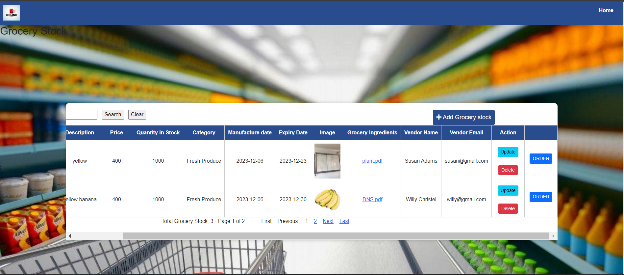
****

The employer will have access to add new jobs by clicking Add Job

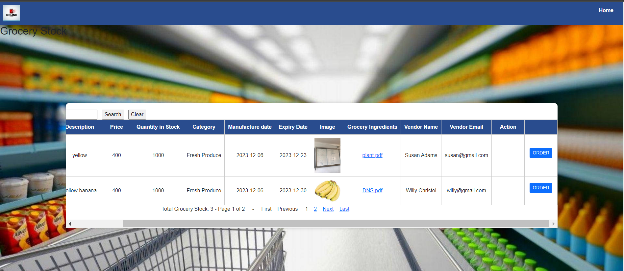


**Listing Available Grocery Stock and Pagenation**

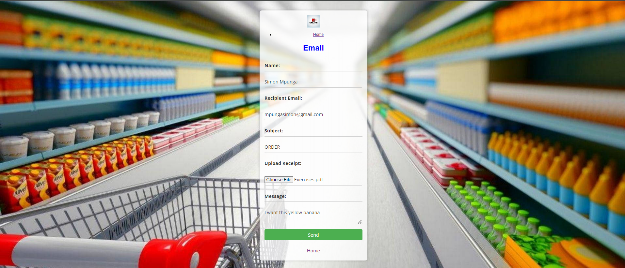
By clicking on Available Jobs you will be able to search for a job by filtering according to the data you have or use a keyword. And also user is able to navigate pages by clicking on next, previous, last and First page. And also user can see the total available jobs



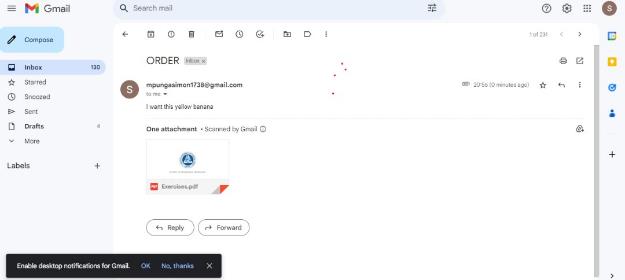
For Spring based roles security user with **ROLE\_BUYER** he/she will not be able to see the Add Job and DELETE AND UPDATE BUTTON



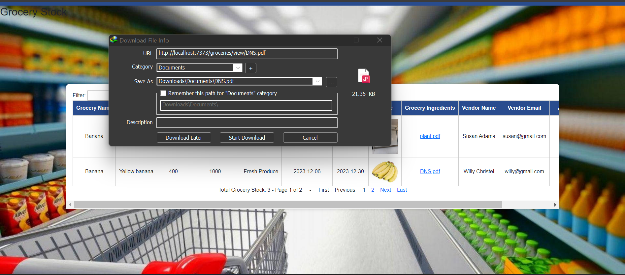
The JOB SEEKER Will be able to apply and send his/her CV via email by clicking apply button and get the email portal



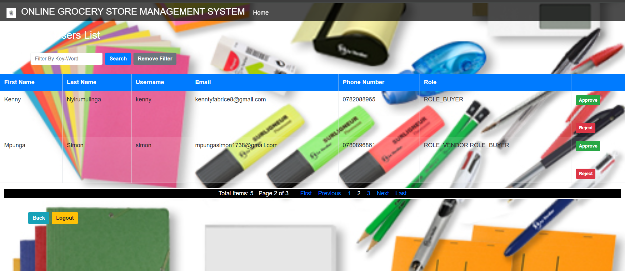
He/she will get notified if the message has reached to the employer via company email



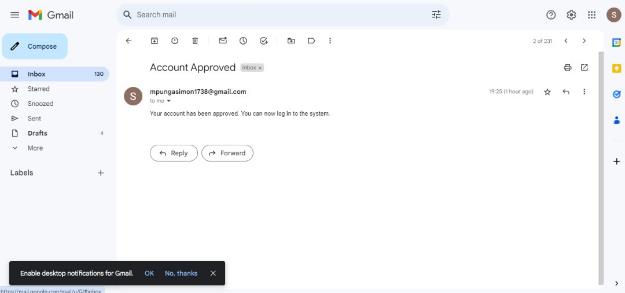
the JOB SEEKER can download the application guidelines that shows the requirement



**User Management**

****

**Approving User**

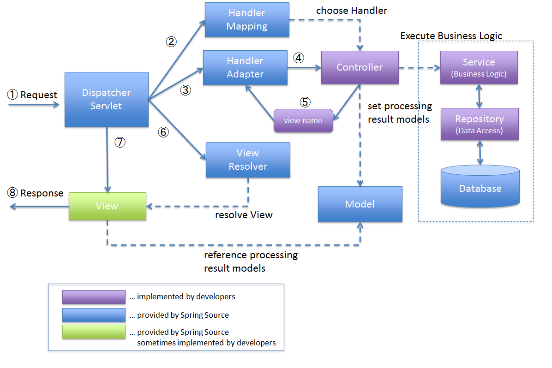


**ERROR HANDLING PAGE**

What if you try to access the wrong URL you will get this error message

**6 . Techinical Documentation**

**Architecture of the application**

****

To implement this architecture, I developed a springboot maven application using spring starter IO. 

* 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-bootstarter-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”, “UserController”, “EmailController”, and “JobController”. The User controller is used to handle requests that are related to signing up new users to the system and setting their roles, the User 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 jobService 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 autoconfiguration.
*  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.

Spring Email : Spring Email is a module within the Spring Framework that provides support for sending emails in Java applications. It simplifies the process of sending emails by offering a high-level abstraction and