Name: axelle isingizwe

**ID: 24271**

**13th ,DECEMBER, 2023**

**WEB TECHNOLOGY FINAPROJECT**

**TOPIC: GET INTERNSHIP**

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

* Develop a get intership platform connecting student and recent graduate.
* Provide an efficient system for centralized internship matching and recruitment.
* Increase visibility for user and widen access to opportunities.

1. **Expected outcomes of the project**

* Offer a diverse range of internship opportunities across the world.
* Implement advanced algorithms for precise intership matching.
* Ensure equal access for all users, overcoming geographical and networking limitations.

1. **Key Functional Requirements**
   * **User Registration and Profiles:** Allow users (studend and graduate, operations managers, employers) to register and create profiles with relevant details.
   * **internship Listings:** Enable employers to post detailed job openings; allow users to search and browse listings.
   * **internship Search and Filtering:** Implement search functionalities based on keywords, location, industry, experience, and more.
   * **Application Submission:** Facilitate direct application submissions, including resumes and cover letters.
   * **Communication and Notifications:** Provide notifications for user actions.
   * **Dashboard:** Include dashboards for administrators to manage user accounts, job listings, and system activities.
   * **Login Page:** Grant access based on user roles.
   * **Forgot and Reset Password:** Allow users to reset passwords through email confirmation.
   * **Pagination:** Implement page navigation for internship listings.
   * **Error Handling:** Display error messages for caught exceptions.
   * **File Upload and Download:** Enable document uploads and downloads for user profiles.
   * **Validation:** Implement error messages for incorrect credentials or data.
2. **Key Non-Functional Requirements**
   * **Reliability:** Ensure 98% accessibility to internship seekers' resumes.
   * **Availability:** Maintain 24/7 system accessibility and responsiveness.
   * **Security:** Restrict access to sensitive information to operations managers and employers.
   * **Authentication:** Enforce strong password requirements for user security.
   * **Data Integrity:** Maintain data integrity with regular database backups.
   * **Usability:** Design a user-friendly and intuitive UI.
   * **Compatibility:** Ensure compatibility with various operating systems and browsers.
   * **Maintainability:** Allocate resources for ongoing maintenance, updates, and bug fixes.
   * **Recoverability:** Implement data recovery mechanisms in case of loss.
   * **Performance:** Load time for the User Interface should not exceed 3 seconds.
   * **Robustness:** Design a robust system capable of handling errors gracefully.

**ii. Project Plan**

* + 1. **SCOPE**

The scope of this project is bounded to the connection between internship seekers and students or graduate in Rwanda. The platform should allow internship seekers and students to register, create profiles, and authenticate themselves to access the platform's features and services. Internship seekers should be able to search for internship listings based on various criteria such as location, internship title, category, experience level, and keywords. The platform should provide filtering and sorting options to help internship seekers find relevant opportunities.

* + 1. **TIMELINE**

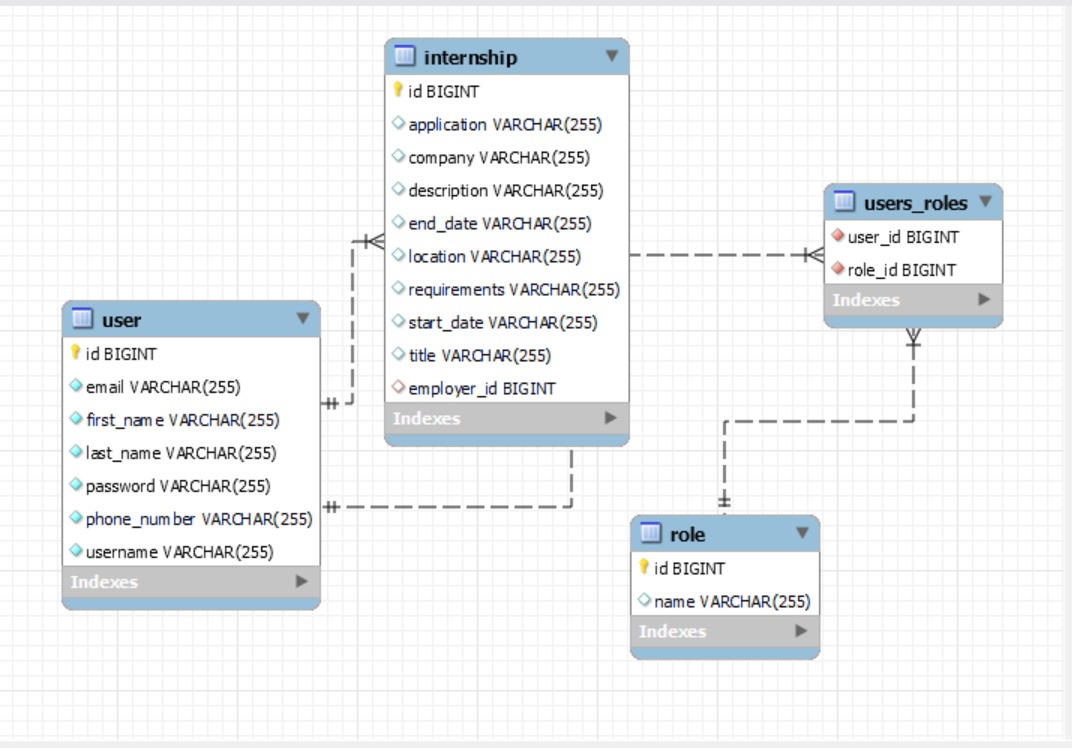
This get internship platform application was developed in a period of one week and a half(12days), requirement gathering took me 1 day, design and prototyping took me 3 days, development and coding took me 6 days, testing the application 1 days and the deployment took me 1 day.

* + 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

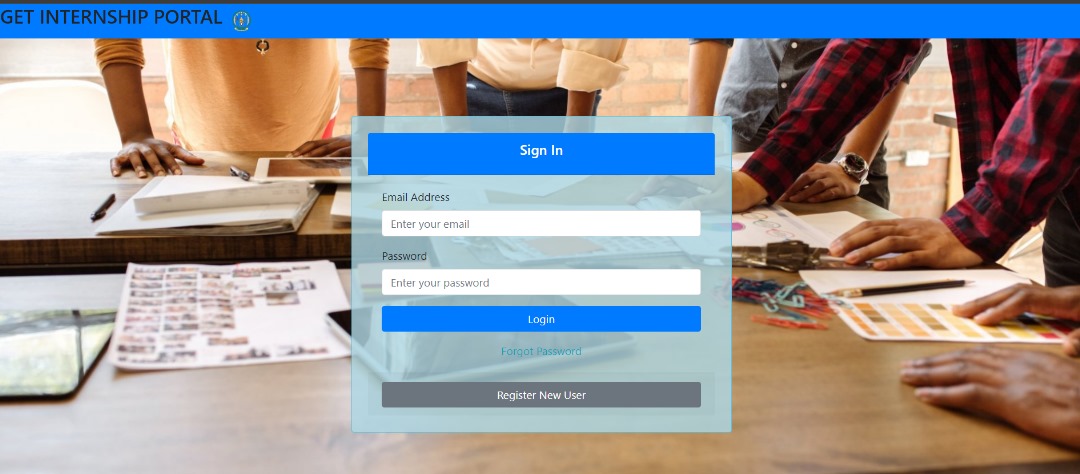
**Web Development Technologies and Tools:** Integrated development environments (IDEs) (e.g., IntelliJ, Visual Studio Code)

**Iv. Database Schema**

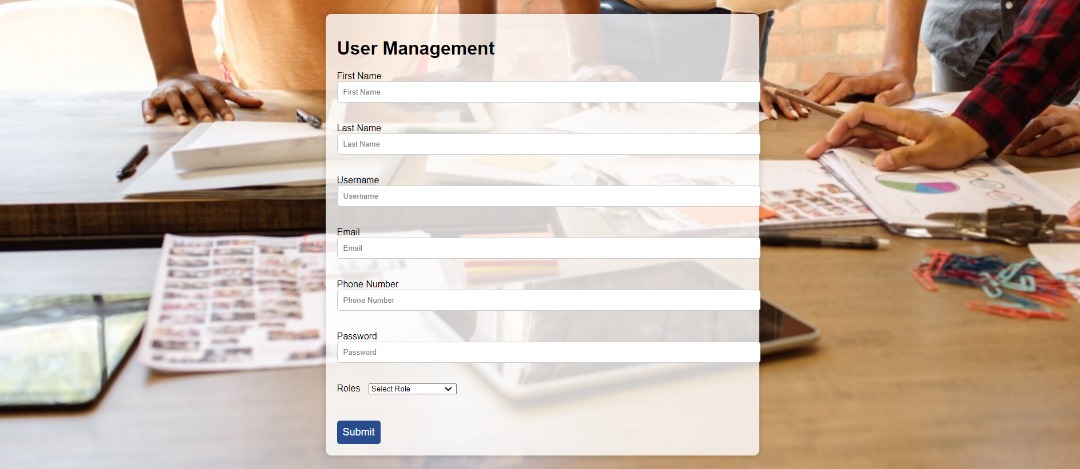
****

**User Documentation To access my App use**

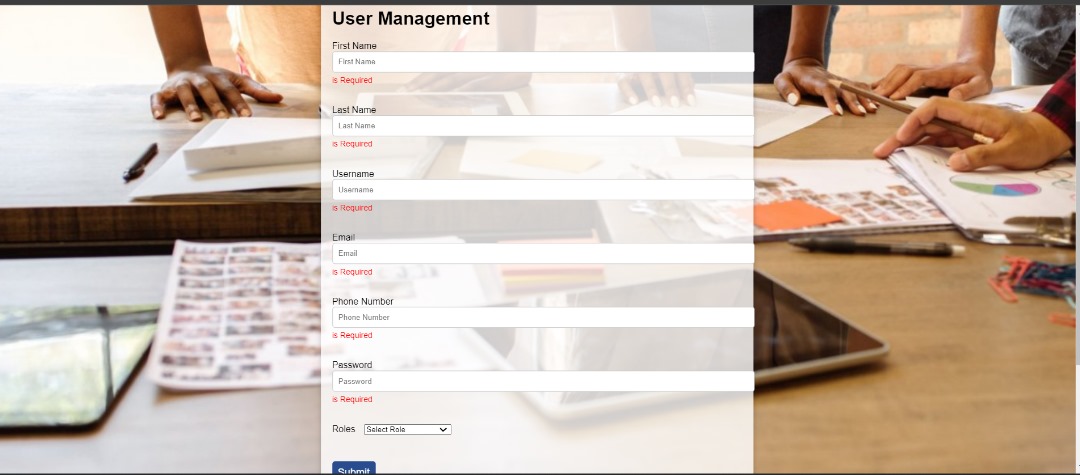
**To access my internship Platform use this link**

**This will take you to the Login Page** 

**Register new user**

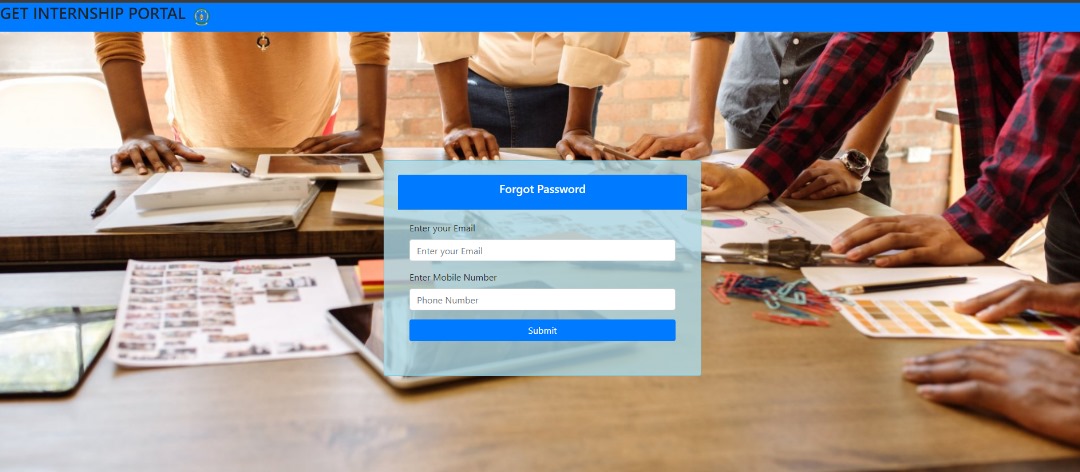
****

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.

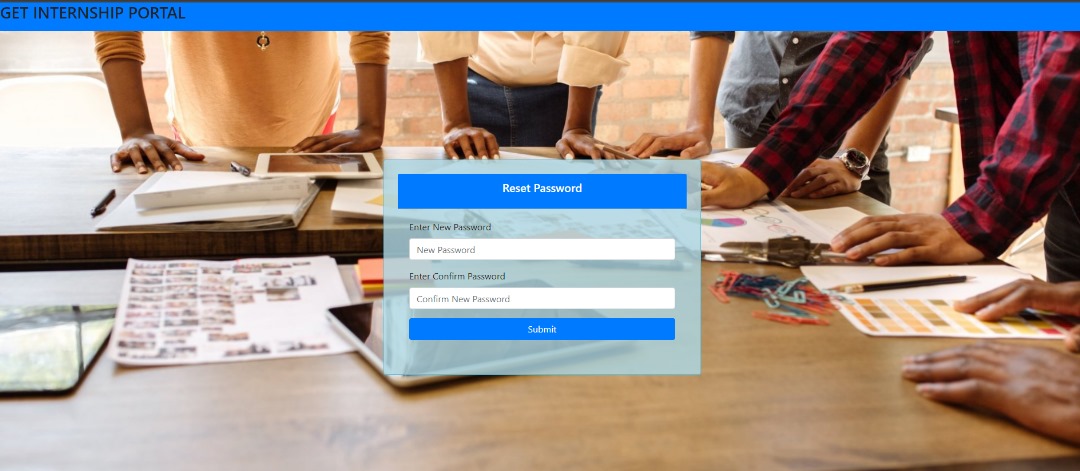
****

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

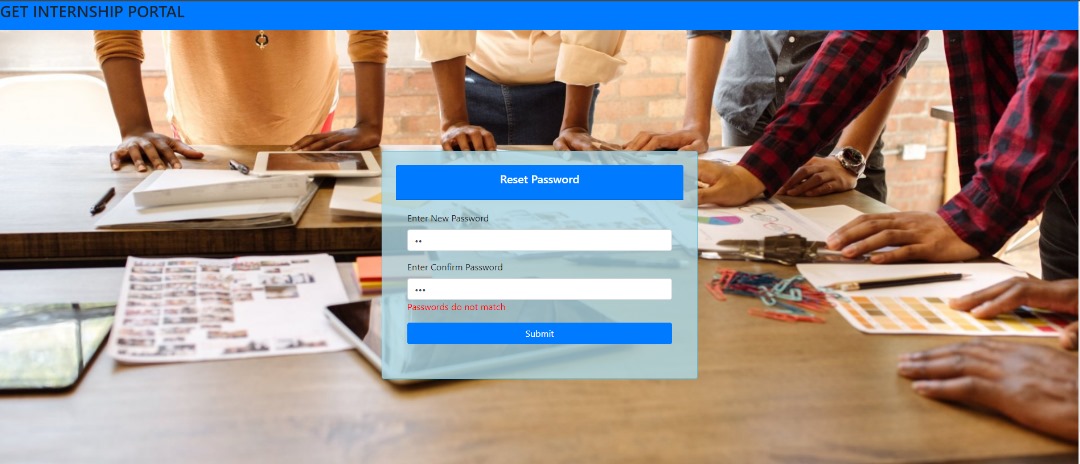
**Click forgot password**

****

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

****

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

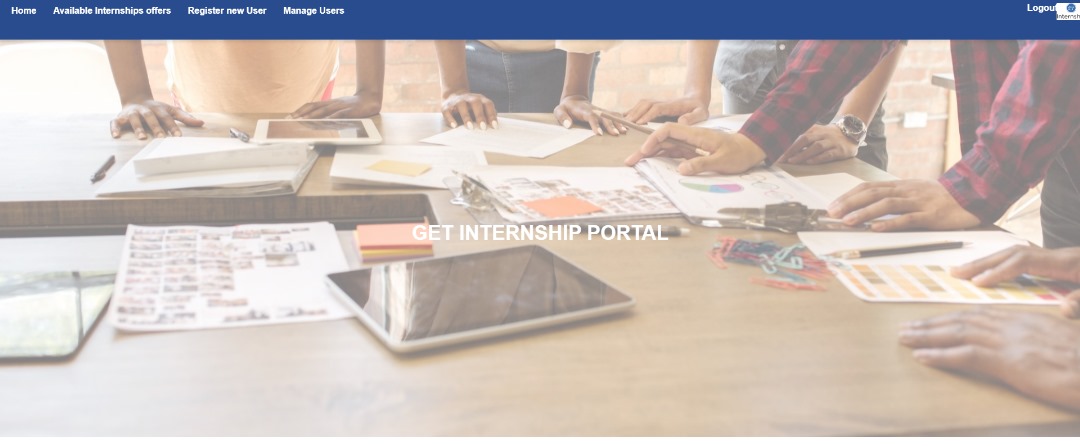
****

And you 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**

****

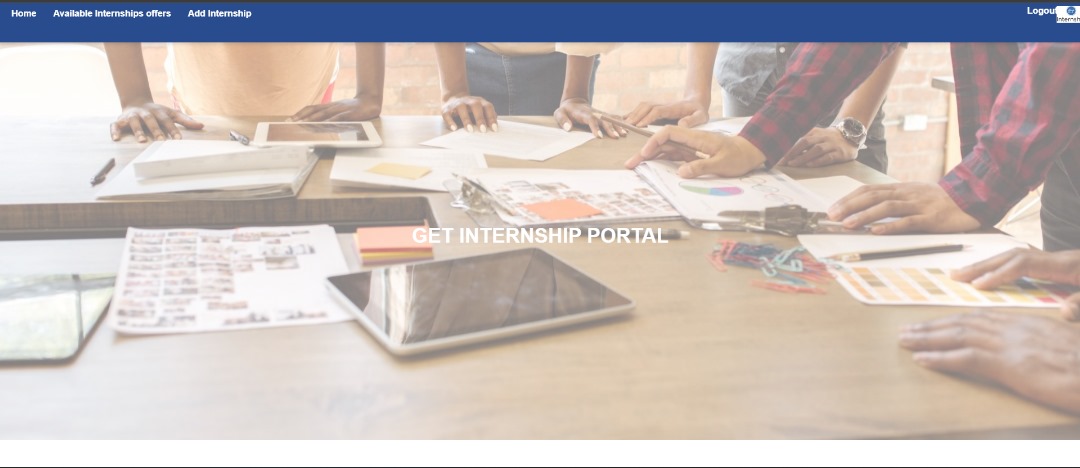
he/she will be able to see available internships, add internship and register new User

**Role user Dashboard**

****

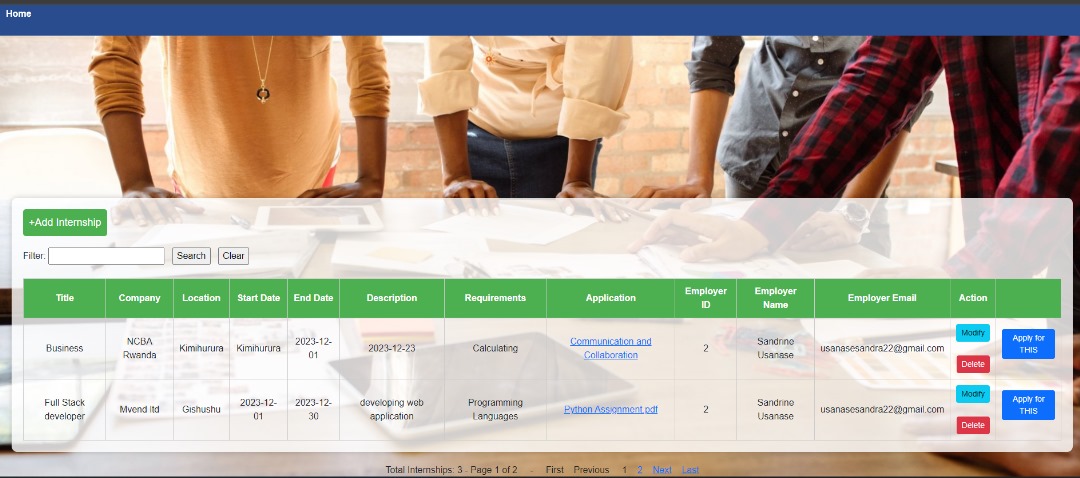
He/she will only have access to view available internships only

**Employer Dashboard**

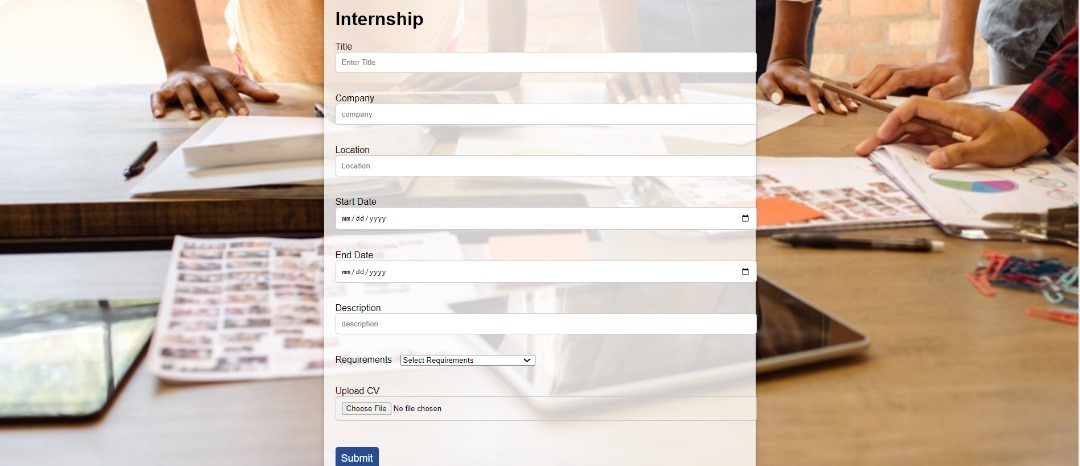


**Listing Available internships**

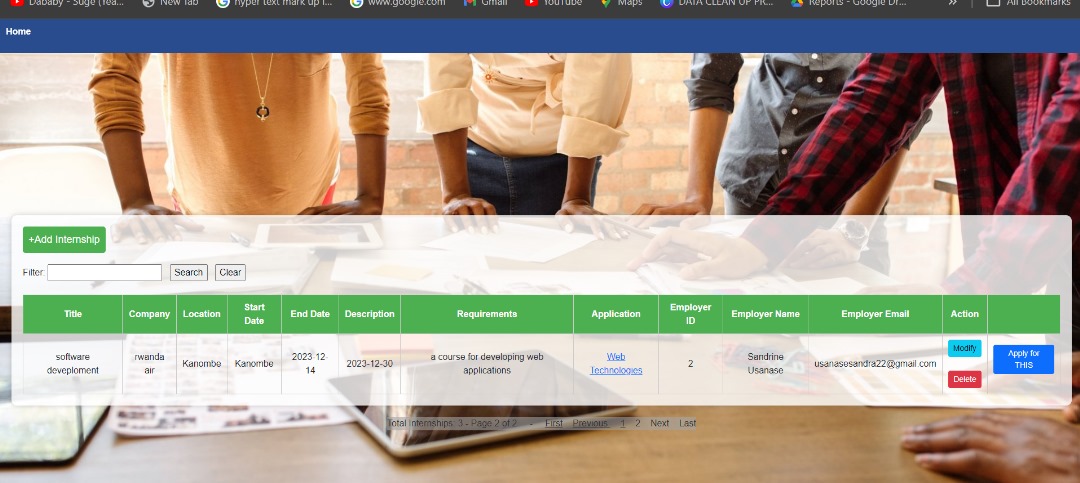
By clicking on Available internships you will be able to search for a internship by filtering according to the data you have or use a keyword. And also user is able to navigate pages by clicking on next, previous, last and First page. And also user can see the total available internships.



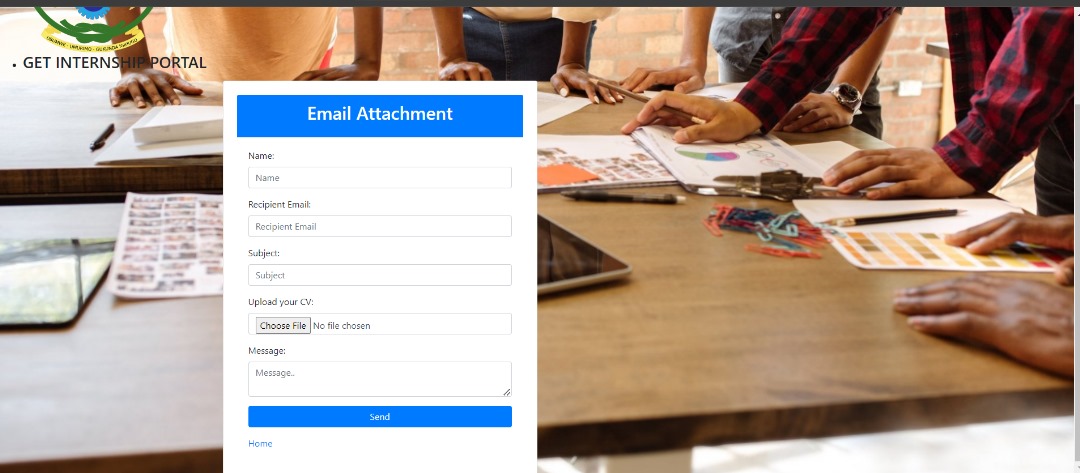
**Add an Internship**

****

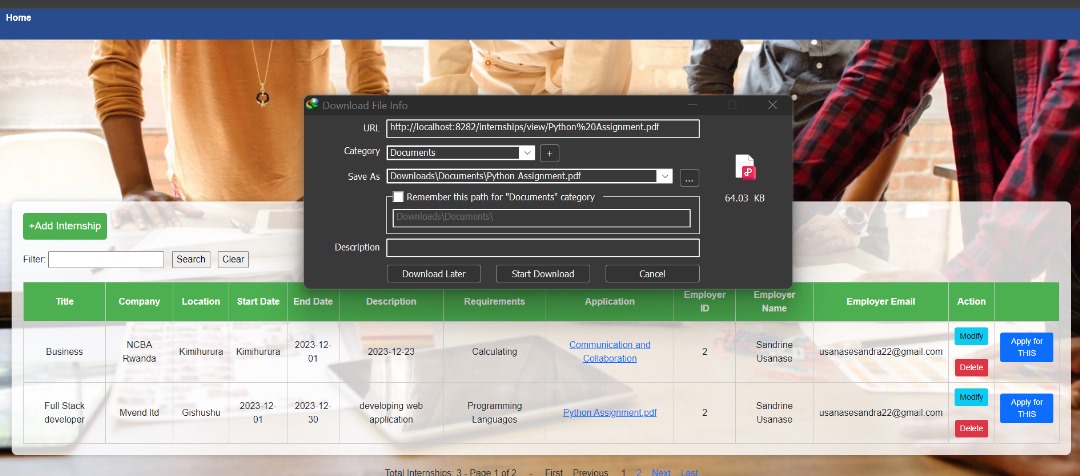
**Pagenation**

****

**User can apply for internship**

****

**Upload and Download the pdf**

****

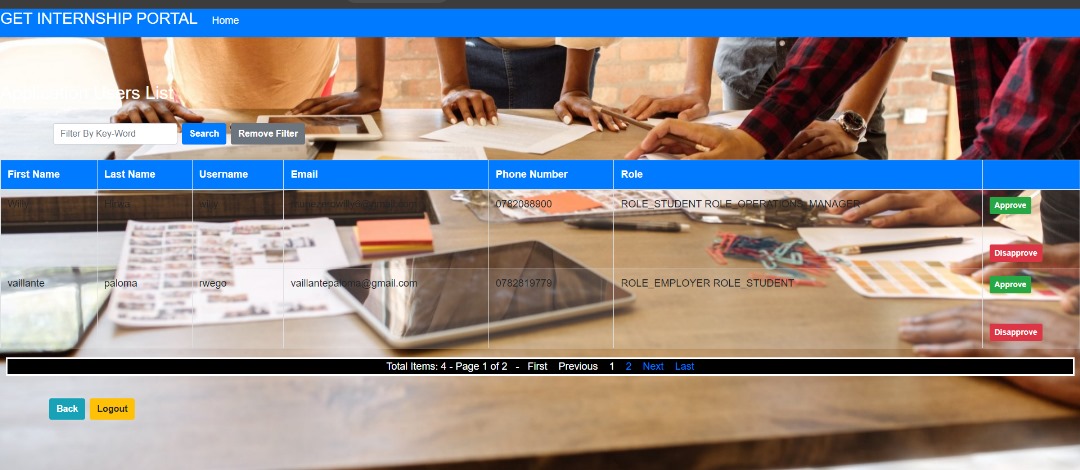
**He/she will get notified if the message has reached to the user via company**

**ERROR HANDLING PAGE**

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



**Manage Users**



**6 . Techinical Documentation**

**Architecture of the application**

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

**1.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.

**2. 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 springboot-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.

**3. ″ 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

**4. 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 internshipService layer that handles the business logic related to construction projects persisted on the application.

**5. 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

**6. 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.

**6.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.

**7. 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

**8. 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.

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

10. **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 integration with various email providers