**Adventist University Of Central Africa**

**Names:** UWIMBABAZI Emmanuella

**Id:** 22949

**Course:** Web Technology

**Project Name: PDF Handling System**

1. **Project description & requirements**

Project Description:

The PDF Handler Service uploads all the documents user wants online to keep them safely either for future use and secure them.

The application should have a user-friendly interface that provides seamless navigation and a secure transaction process.

Requirements:

1.User Registration and Authentication:

Users should be able to create an account, log in.

1. PDF Handling System Creation:

Admin will be able to create events by providing event details such as name, description, pdf file and actions.

3.PDF Handling System Listing and Search:

Event organizers should be able to create and manage event listings, including details like event name, date, time, location, ticket prices.

Users should be able to search for pdf based on a criteria such as name.

4.PDF Handling System Management:

Users should be able to view their uploaded document.

PDF Handling System organizers should have access to a dashboard to track documents, view uploaded document, and generate reports.

7.Admin Panel:

Create an admin panel to manage document uploaded, with their description and disputes.

8.Documentation and Testing:

Perform comprehensive testing to ensure the application functions correctly and is free from bugs or vulnerabilities.

9.Future Scalability:

Design the application with scalability in mind, allowing for the addition of new features and accommodating a growing user base.

**ii.Project Plan**

**Project Plan**

Scope:

1. Analysis and requirements gathering: Gather detailed requirements from stakeholders and define the scope of the project.

2. System design: Design the architecture and database structure of the web application.

3. User interface design: Create user-friendly and responsive interfaces for both admin and users.

4. PDF Handling System management: Implement features for event creation, editing, and deletion by admin.

5. Users management: Develop functionalities for Name, Description and Select PDF file.

6. Reports and analytics: Implement reporting features to provide admin with insights into uploaded file and Names.

7. User management: Develop user registration, login, and account management features for Admin and Users.

8. Testing and quality assurance: Thoroughly test the application to ensure functionality, security, and reliability.

10. Deployment: Deploy the application to a production environment.

Timeline:

1. Day 1: Analysis and requirements gathering.

2. Day 2: System design and database structure.

3. Day 3: User interface design.

4. Day 4-9: PDF Handling System implementation.

5. Day 10-11: Admin management implementation.

6. Day 12-13: User management implementation.

7. Day 14-17: API Testing

8.Day 18: User interface Validation

Resources:

1. Project Manager: Responsible for overall project coordination, task assignment, and timeline management.

2. Developers: Front-end and back-end developers responsible for implementing various features of the web application.

3. UI/UX Designer: Responsible for designing user-friendly and responsive interfaces.

4. QA/Testers: Responsible for conducting thorough testing of the application to ensure functionality and reliability.

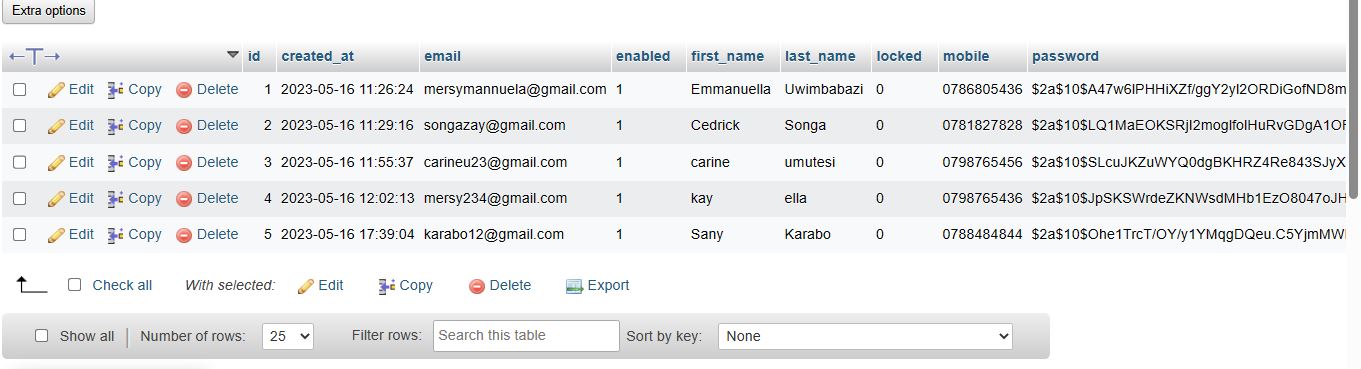
5. Database Administrator: Responsible for designing and managing the database structure.

6. Technical Writer: Responsible for documenting the project requirements, user guides, and technical specifications.

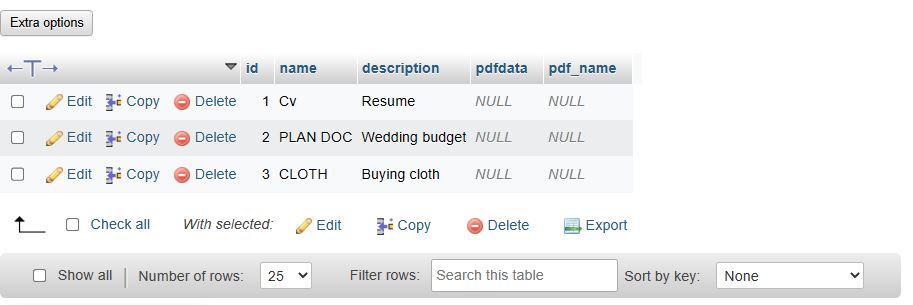
7. Deployment Specialist: Responsible for deploying the application to a production environment and ensuring proper configuration.

**Iv Database Schema**

**ADMIN**

****

**USERS**

****

**v. User Documentation**

User Guide: Event Ticket Management Web Application

Welcome to the User Guide for the Event Ticket Management web application. This guide will walk you through the features and functionalities of the application, providing step-by-step instructions to help you effectively use the system. Let's get started!

Table of Contents:

1. Creating an Account

2. Logging In

3. Event Creation and Management

4. Event Ticket Creation and Management

5.Client View Event created

**1. Creating an Account:**

To start using the PDF Handling System application, you need to create an account. Follow these steps to create your account:

1. Visit the application's website.
2. Click on the "Register" button.
3. Fill in the required information, such First name, Last name, email address, phone number and password.
4. Submit the form.
5. Congratulations! You have successfully created your account, Immediately you’ll be directed to Login page.

**2. LOG IN :**

Once you have an account, follow these steps to log in to the application:

a. Visit the application's website.

b. Click on the "Log In" button.

c. Enter your email address and password.

d. Click on the "Log In" button.

e. You are now logged in to the application and ready to explore its features.

**3.PDF Handling System(USER):**

As a User to upload a pdf to the system

* Visit the application's website
* Insert the name of the document to be uploaded
* Add the ”Description”.
* Select or Choose “PDF File”
* Click on “Upload” button

**4.PDF Handling System(ADMIN):**

As an Admin to access and view upload the pdf users

* Visit the application's website
* Log in into the account
* Click on “View Uploaded PDF”
* You can download also.

**vi. Technical Documentation**

**Introduction:**

The Technical Documentation for the Event Ticket Management web application provides detailed information about the application's architecture, components, dependencies, and configuration.

Table of Contents:

1. System Overview

2. Architecture

3. Technologies Used

4. List of Dependency Used

7. APIs and Integration

8. Security Considerations

9. Testing

10. Deployment

11. Monitoring and Maintenance

12. Troubleshooting

13. Release Notes

**1. System Overview:**

The PDF Handling System web application is built using Spring Boot, a powerful Java framework that simplifies the development of robust and scalable applications. Spring Boot provides a comprehensive ecosystem for building web applications with features like dependency injection, MVC framework, and seamless integration with various libraries and tools.

**2. Architecture:**

The application follows a layered architecture, with the presentation layer implemented using Spring MVC and Thymeleaf for server-side rendering. Spring Boot acts as the back-end application framework, handling business logic and data access. The application leverages Spring Data JPA for database interactions and Spring Security for authentication and authorization.

1. **Technologies Used:**

- Backend: Spring Boot (Java)

- Frontend: HTML5, CSS3 and Bootstrap

- Database: MySQL

- Testing: Spring Test

1. **List of Dependency Used:**

1. Spring Boot Starter Web: Provides the necessary dependencies for building web applications with Spring Boot, including an embedded web server and the Spring MVC framework.

2. Spring Boot Starter Thymeleaf: Enables server-side rendering using Thymeleaf, a popular Java-based template engine for web applications.

3. Spring Boot Starter Data JPA: Provides support for data access and persistence using Spring Data JPA, including database connectivity, entity management, and repository interfaces.

4. Spring Boot Starter Security: Facilitates authentication and authorization in the application using Spring Security, enabling secure access to resources and protecting against common security threats.

5. Spring Boot Starter Validation: Offers validation support by integrating the Hibernate Validator, allowing validation of form inputs and request payloads.

6. MySQL Connector/J: Enables connectivity to MySQL database by providing the JDBC driver.

7.Spring Data JPA to provide object-relational mapping capabilities and database persistence.

8. Spring Boot Starter Test: Provides dependencies for testing Spring Boot applications, including JUnit and Mockito for unit testing and Spring Test for integration testing.

9. Thymeleaf Extras Spring Security: Integrates Spring Security with Thymeleaf, enabling the use of security-related tags and expressions in the HTML templates.

10. Lombok: Simplifies Java code development by reducing boilerplate code through annotations, improving code readability and maintainability.

1. **Security Considerations:**

User Authentication: we have Implemented a secure user authentication mechanism to ensure that only authorized users can access the application. We have also Utilized strong password policies, such as enforcing 8 minimum password length.

**GitHub link: https://github.com/sanymersy/pdf-handler-service-16-**