<u>Adventist University Of Central Africa</u>

Names: Harerimana Umuhire Clement

Id:23388

Course: Web Technology

Project Name: Event Ticket Web Application

I. Project description & requirements

Project Description:

The Event Ticket Management web application is a platform designed to streamline the process of managing and selling tickets for events. This application will enable event organizers to create and manage events, sell tickets, track attendance, and generate reports. The application will also provide a user-friendly interface for ticket buyers to purchase tickets, view event details, and manage their tickets.

The application should have a user-friendly interface that provides seamless navigation and a secure transaction process. It should also incorporate features to help event organizers manage their ticket inventory, track sales, and generate reports.

Requirements:

1.User Registration and Authentication:

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

2. Event Creation:

Admin will be able to create events by providing event details such as name, date, time, location, and ticket prices.

3.Event 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 events based on different criteria such as location, date, etc.

4.Ticket Purchasing:

Users should be able to browse events and purchase tickets online.
Implement a secure payment gateway integration to facilitate transactions.
Provide users with electronic tickets that can be downloaded or sent via email.

5.Ticket Selling:

Users should have the option to sell their purchased tickets if they are unable to attend an event.

Implement a feature to list tickets for sale, specify the selling price, and manage the selling process.

6.Ticket Management:

Users should be able to view their purchased tickets.
Event organizers should have access to a dashboard to track ticket sales, view attendee information, and generate reports.

7.Admin Panel:

Create an admin panel to manage events listings, with their tickets and disputes or issues related to transactions.
Admins should have the ability to moderate event listings and create Tickets accordingly.

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 event organizers and ticket buyers.
4. Event management: Implement features for event creation, editing, and deletion by event organizers.
5. Ticket management: Develop functionalities for ticket sales, ticket scanning, and ticket management for ticket buyers.
6. Reports and analytics: Implement reporting features to provide event organizers with insights into ticket sales, attendance, and revenue.
7. User management: Develop user registration, login, and account management features for event organizers(Admin) and ticket buyers.
8. Payment integration: Integrate payment gateways to enable secure online payment processing for ticket purchases.
9. 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: Event management implementation.
5. Day 10-11: Ticket management implementation.
6. Day 12-13: User management implementation.
7. Day 14-17: API Testing
8.Day 18:User interface Validation

Resources:
List of Resources Used to during development of Event Ticket Web Application:

1. Official Spring Boot Documentation: The official documentation provided by the Spring team is an excellent resource to understand the concepts, features, and usage of Spring Boot. It includes comprehensive guides, tutorials, and references that cover various aspects of Spring Boot development. Visit: https://docs.spring.io/spring-boot/docs/

2. Spring Boot Reference Guide: The Spring Boot Reference Guide provides in-depth information about the various Spring Boot features, configurations, and customization options. It covers topics such as application properties, dependency management, data access, security, testing, and deployment. Visit: https://docs.spring.io/spring-boot/docs/current/reference/html/

3. Spring Boot Starters GitHub Repository: The Spring Boot Starters repository on GitHub contains a collection of starter dependencies that help bootstrap the development of Spring Boot applications. Exploring the repository can provide insights into the available starters and their configurations. Visit: https://github.com/spring-projects/spring-boot-starters

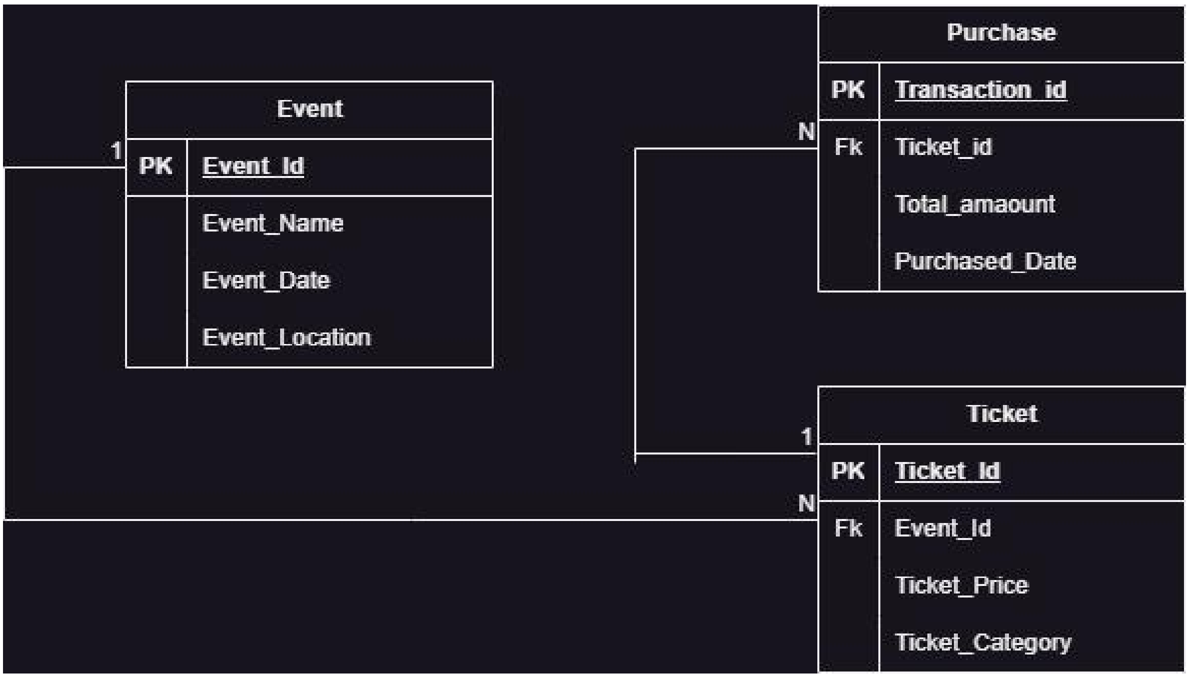
4. Spring Boot Examples GitHub Repository: The official Spring Boot Examples repository on GitHub provides a variety of sample applications demonstrating different aspects of Spring Boot development. These examples cover topics like REST APIs, database integration, security, testing, and more. Visit: https://github.com/spring-projects/spring-boot/tree/main/spring-boot-samples

5. Spring Boot Community: The Spring Boot community is active and vibrant, with many developers sharing their experiences, insights, and best practices. Joining the community can provide access to forums, discussion groups, and Q&A platforms where you can ask questions, seek guidance, and learn from others' experiences. The official Spring Boot community page provides links to various community resources. Visit: https://spring.io/community

6. Spring Boot Blogs and Articles: Many technology blogs and websites regularly publish articles and blog posts related to Spring Boot. Sites like Baeldung, DZone, and the official Spring blog provide valuable insights, tutorials, and best practices for Spring Boot development.

7. Spring Boot YouTube Channels and Videos: Several YouTube channels specialize in Spring Boot tutorials and walkthroughs. Channels like "Java Brains" and "Spring Framework Guru" offer video tutorials that cover various Spring Boot topics, from basic concepts to advanced techniques.

Iv Database Schema



Event	
PK	Event_Id
	Event_Name Event_Date Event_Location

Purchase	
PK	Transaction_id
Fk	Ticket_id Total_amaount Purchased_Date

Ticket	
PK	Ticket_Id
Fk	Event_Id Ticket_Price Ticket_Category

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 Event Ticket Management application, you need to create an account. Follow these steps to create your account:

a. Visit the application's website.
b. Click on the "Sign Up" button.
c. Fill in the required information, such email address and password.
d. Submit the form and verify your email address, if required.
e. Congratulations! You have successfully created your account.

2. Logging 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. Event Creation and Management:

As an Admin, you can create and manage events. Here's how:

a. Log in to the application.
b. Click on the "Create Event" button on Upper Menu.
c. Fill in the event details, such as the event Id, date, Event Name, location, and location.
d. Click on the "Save" or "Register" button.
e. You have successfully created an event. You can create Ticket for an Event.
f. to Set the event tickets and click on Ticket under Action in a Table View.
g. You have successfully created an event's Ticket. You can manage your tickets. Under the action in Ticket Table View.

4. Ticket Purchase:

As a ticket buyer, you can purchase tickets for events. Follow these steps:

a. Log in to the application.
b. Browse the available events or search for a specific event.
c. Select the desired event and click on the "Buy Tickets" or "Purchase" button.
d. Choose the ticket type and quantity.
e. Proceed to the payment gateway and complete the transaction.
f. After successful payment, you will receive your tickets via email.
g. Access your purchased tickets in the "My Tickets" or "Tickets" section.

N.B The last functionality is not yet implemented.

vi. Technical Documentations

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
5. Security Considerations

1. System Overview:

The Event Ticket Management 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.

3. Technologies Used:

- Backend: Spring Boot (Java)
- Frontend: HTML5, CSS3 and BootStrap
- Database: MySQL
- Testing: Spring Test

4. 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.

5. 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.