DAYANANDA SAGAR UNIVERSITY

Devarakaggalahalli, Harohalli,
Kanakapura Road, Ramanagara- 562112, Karnataka, India

SCHOOL OF
ENGINEERING

Bachelor of Technology in
COMPUTER SCIENCE AND ENGINEERING

Spring Boot

Event Management System

By

VISHAL S - ENG22CS0505

PIHU MITTAL - ENG22CS0391

Date of Submission: 20/12/2024

Under the supervision of

Dr. S. Gokulakrishnan

Assistant Professor

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING,

SCHOOL OF ENGINEERING DAYANANDA SAGAR UNIVERSITY

(2024-2025)

ACKNOWLEDGEMENT

We are very much thankful to our guide Dr. S. Gokulakrishnan, Assistant Professor, Dept of CSE for providing help and suggestions in completion of this project successfully.

The satisfaction that accompanies the successful completion of a task would be incomplete without the mention of the people who made it possible and whose constant guidance and encouragement crown all the efforts with success.

We are especially thankful to the Dean, School of Engineering Dr. Udaya Kumar Reddy K R, Dept of CSE, Chairman Dr. Girisha G S, for providing necessary departmental facilities, moral support and encouragement.

TABLE OF CONTENTS

1. INTRODUCTION

This project involves the development of an event management system designed for administrators and customers to manage and book events. The system integrates features such as user authentication, event categorization, event booking, and administrative controls. By using Spring Boot for backend services and React for the frontend, this system allows both administrators and customers to efficiently interact with event data in a secure and user-friendly environment. The system also aims to provide real-time updates on available events and ticket prices.

1.1 Aim of the project

This project aims to develop a comprehensive event management system that allows customers to register, log in, browse, and book events while enabling administrators to efficiently manage events, categories, and customer bookings. It will feature a dynamic and responsive interface for event discovery and bookings, ensuring role-based authentication for both admins and customers.

The system will enable secure user authentication, including distinct roles for admins and customers, and provide a user-friendly interface where customers can search, view, and book events. Administrators will be empowered to manage events, categories, and users. The project will also implement an event booking system with payment functionality for customers and design a robust backend service with efficient database management using MySQL, alongside a secure RESTful API architecture.

1.2 Scope of the Project

The scope of the project includes:

- Creating and managing user accounts for both admins and customers.
- Developing functionality for admins to manage events, categories, and user roles.

- Enabling customers to search, view, and book events with payment processing.
- Integrating a MySQL database to handle user and event data efficiently. However, the scope does not include complex machine learning functionalities or large-scale deployment considerations.

The project will not include:

- Advanced payment gateway integration.
- Mobile application development.
- Analytics or reporting on event performance.

1.3 Significance

This project is significant because it provides a real-world application for managing events in a dynamic and interactive environment. It offers both administrative control and user-friendly interfaces, bridging the gap between event organizers and participants. The integration of role-based access control ensures that the system remains secure, while the flexibility of the frontend design allows scalability for future features or updates.

2. LITERATURE REVIEW

2.1 Summary of Existing Research:

Existing research in the field of event management systems primarily focuses on the automation of event registrations, ticketing, and user management. Most systems use basic CRUD operations and rely on user authentication methods, which are well-studied in secure web development practices. Studies have also explored the integration of payment gateways into event booking systems to ensure secure transactions.

2.2 Key Findings:

- Many event management systems focus on simplifying ticket sales and event promotions, but few provide extensive admin controls like user role differentiation or category management.
- Research suggests that seamless integration of databases with event systems enhances the user experience by allowing real-time data retrieval and updates.
- Security measures, particularly in multi-role user systems (admin vs. customer), are crucial for preventing unauthorized access.

2.3 Research Gap:

While there is existing research on the development of event management systems, few studies focus on the holistic integration of user roles, complex event categorization, and payment gateways in a way that is both user-friendly and secure. This project addresses this gap by focusing on:

- Implementing advanced admin controls and user management.
- Providing a seamless experience for customers to search and book events securely.
- Integrating a payment system for real-time event booking

3. METHODOLOGY

3.1 Research Design:

This project follows a practical, experimental approach where the focus is on creating a fully functional prototype. The research combines both qualitative (user feedback, usability testing) and quantitative (performance data, error rates) research methods to evaluate system performance and usability.

3.2 Materials and Tools:

- Backend: Spring Boot, MySQL, Java
- Frontend: React, TypeScript, CSS
- Payment Gateway: Custom implementation for event booking
- Development Environment: Visual Studio Code, IntelliJ IDEA, Postman for API testing
- Database: MySQL (with MySQL Connector)

3.3 Procedure:

1. Sample Selection: The system is designed to be used by two types of users: administrators and customers.
2. Data Collection Methods: Event data (category, name, venue) is gathered through forms in the admin panel, while customer data (name, email, ticket booking) is collected via registration and event booking forms.
3. Analysis Techniques: The system's effectiveness will be evaluated based on its ability to handle multiple events, user roles, and bookings simultaneousl

4. Implementation

4.1 Process:

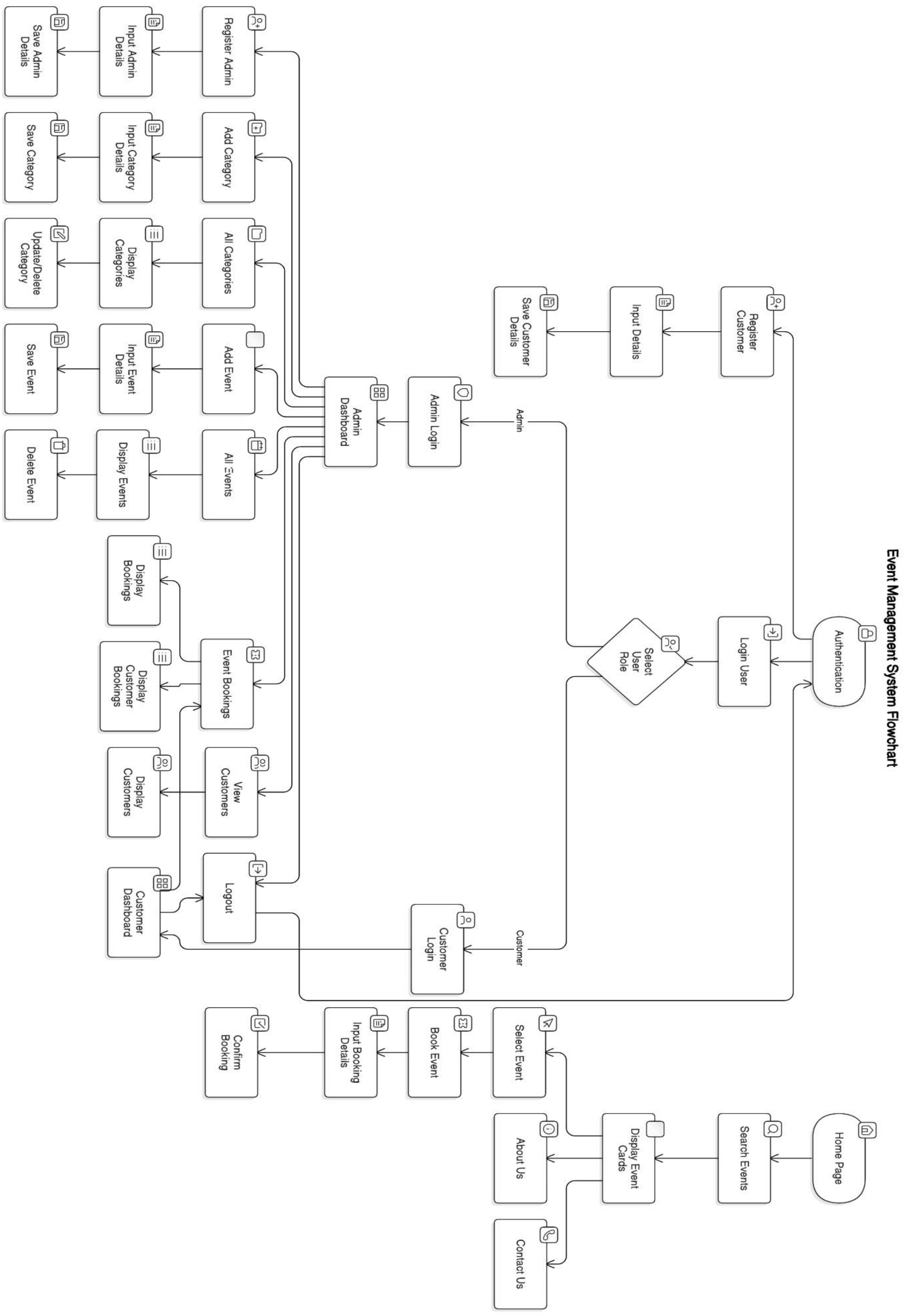
1. **Authentication Implementation**: User authentication was designed with role-based access control using Spring Security. Admins and customers have distinct access privileges.

2. **Event Management**: Admins can add and manage events, including category selection, ticket pricing, and event images. This system is connected to a MySQL database to store and retrieve data.

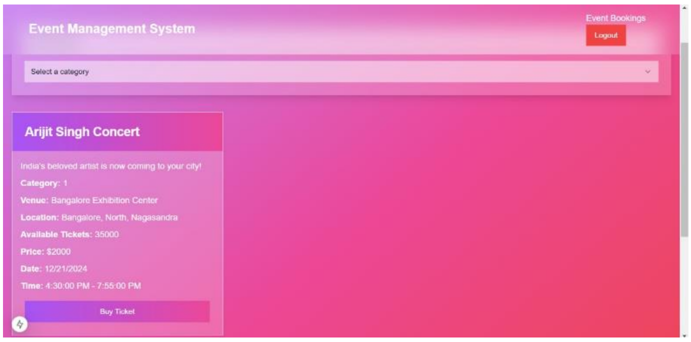
3. **Booking Process**: For customers, the event booking process involves selecting an event, entering personal details, and receiving a confirmation.

4. **Challenges**: Ensuring secure payment processing and seamless role-based access control were primary challenges. These were addressed by integrating secure authentication methods and testing with mock payment data

4.2 Flowchart



4.3 Screenshots



Event Management System

Event Bookings

Arijit Singh Concert

India's beloved artist is now coming to your city!

Category: 1

Venue: Bangalore Exhibition Center

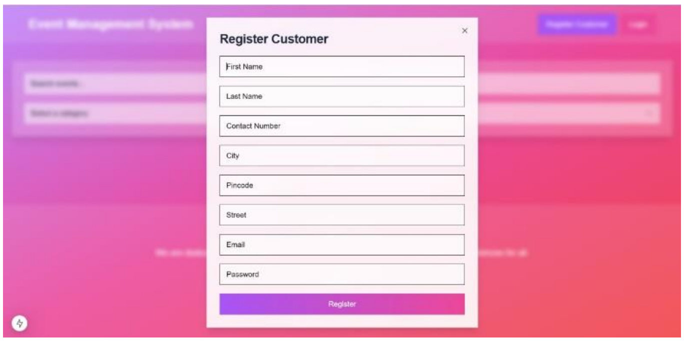
Location: Bangalore, North, Nagasandra

Available Tickets: 35000

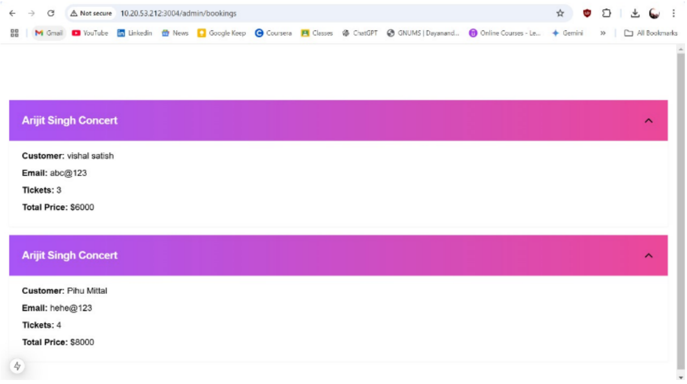
Price: $2000

Date: 12/21/2024

Time: 4:30:00 PM - 7:55:00 PM



Register Customer



Not secure 10.20.53.212:3004/admin/bookings

Arijit Singh Concert

Customer: vishal satish

Email: abc@123

Tickets: 3

Total Price: $6000

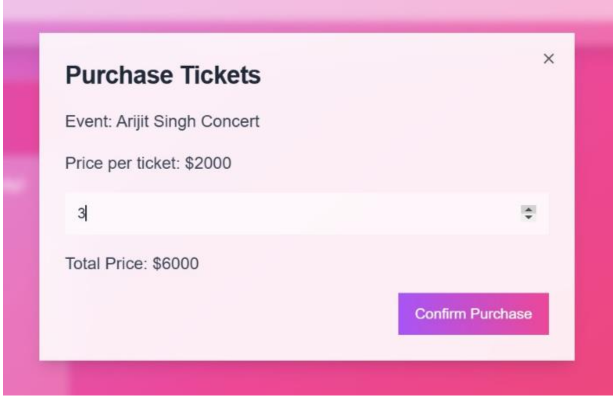
Arijit Singh Concert

Customer: Pihu Mittal

Email: hehe@123

Tickets: 4

Total Price: $8000



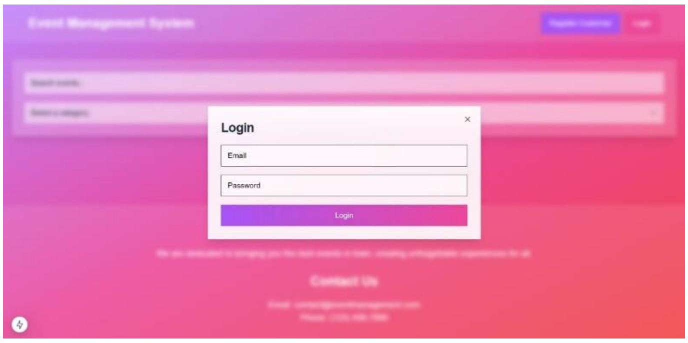
Purchase Tickets

Event: Arijit Singh Concert

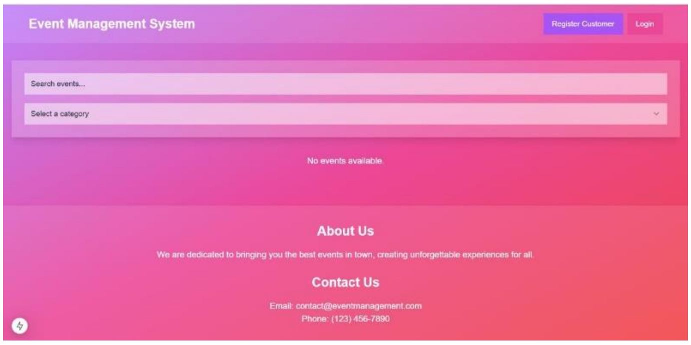
Price per ticket: $2000

3

Total Price: $6000



Login



Event Management System

No events available.

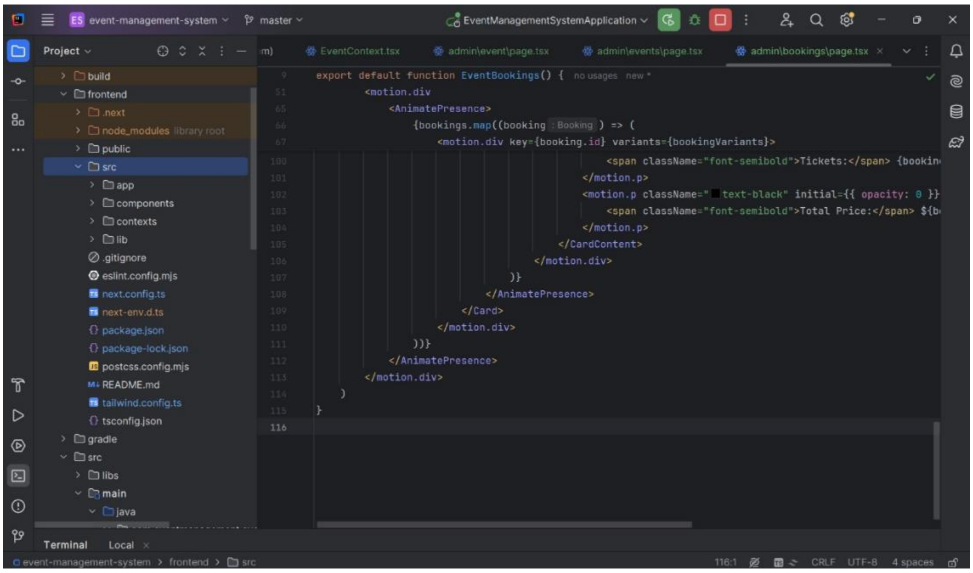
About Us

We are dedicated to bringing you the best events in town, creating unforgettable experiences for all.

Contact Us

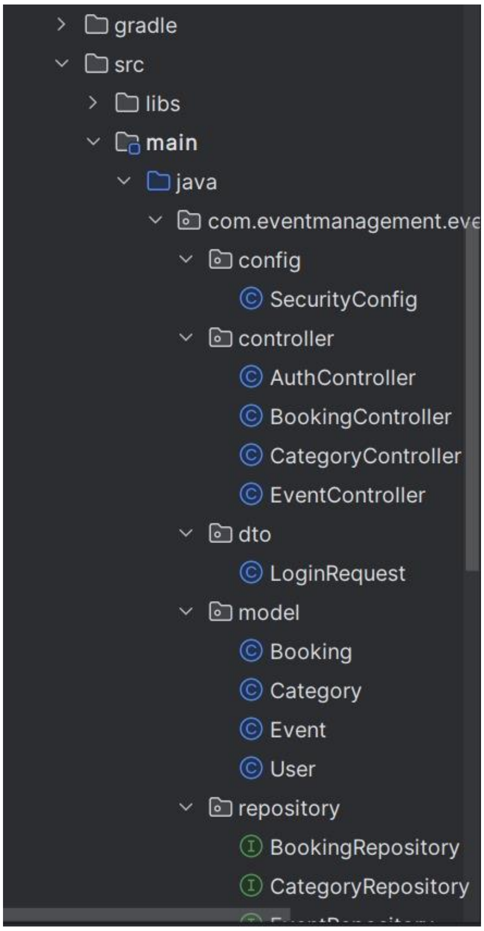
Email: contact@eventmanagement.com
Phone: (123) 456-7890

4.4 Project Structure in IntelliJ IDE

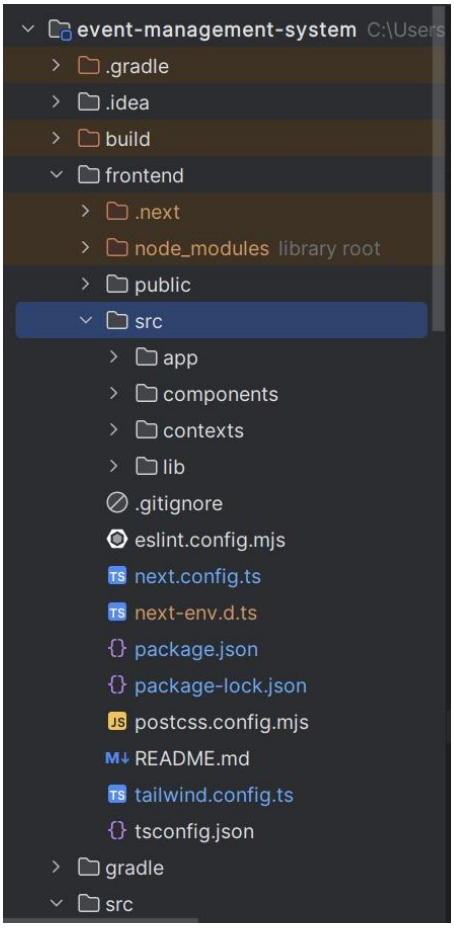


```
build
frontend
    .next
    node_modules library root
    public
    src
        app
        components
        contexts
        lib
    .gitignore
    eslint.config.mjs
    next.config.ts
    next-env.d.ts
    package.json
    package-lock.json
    postcss.config.mjs
    README.md
    tailwind.config.ts
    tsconfig.json
gradle
src
    libs
    main
        java
```

```
export default function EventBookings() {  no usages  new *
        <motion.div
            <AnimatePresence>
                {bookings.map((booking : Booking ) => (
                    <motion.div key={booking.id} variants={bookingVariants}>
                                    <span className="font-semibold">Tickets:</span> {bookin
                                </motion.p>
                                <motion.p className=" text-black" initial={{ opacity: 0 }}
                                    <span className="font-semibold">Total Price:</span> ${b
                                </motion.p>
                            </CardContent>
                        </motion.div>
                    ))}
                </AnimatePresence>
            </Card>
        </motion.div>
    ))}
    </AnimatePresence>
</motion.div>
    )
}
```



```
gradle
src
    libs
    main
        java
            com.eventmanagement.eve
                config
                    SecurityConfig
                controller
                    AuthController
                    BookingController
                    CategoryController
                    EventController
                dto
                    LoginRequest
                model
                    Booking
                    Category
                    Event
                    User
                repository
                    BookingRepository
                    CategoryRepository
```



```
event-management-system  C:\Users
    .gradle
    .idea
    build
    frontend
        .next
        node_modules  library root
        public
        src
            app
            components
            contexts
            lib
        .gitignore
        eslint.config.mjs
        next.config.ts
        next-env.d.ts
        package.json
        package-lock.json
        postcss.config.mjs
        README.md
        tailwind.config.ts
        tsconfig.json
    gradle
    src
```

5. Results and Discussions

5.1 Data Presentation:

Tables and graphs representing the number of events created, number of users registered, and successful bookings.

5.2 Analysis:

- The implementation of event categorization allowed for easier navigation and a better user experience.
- Role-based access control ensured secure management of event data by admins while providing a restricted, customer-friendly interface.
- The booking system worked as expected, with real-time payment processing and ticket allocation.

5.3 Comparison:

Compared to other similar event management systems, this project offers a more robust admin interface and a more secure, flexible payment system.

5.4 Limitations:

- Limited scalability for handling extremely high traffic.
- Payment integration was based on a simplified mock-up, and real-world testing is yet to be done.

6. Conclusion

6.1 Summary of Findings:

The project successfully developed an event management system with features for both admins and customers. The system supports secure user authentication, dynamic event and category management, and a seamless event booking process.

6.2 Implications:

This system can be expanded for use in real-world event management by adding more categories, user roles, and integrating real payment gateways.

6.3 Recommendations:

Future work should focus on:

- Expanding the database to handle larger numbers of users and events.
- Integrating more advanced features such as event promotions and user reviews.
- Implementing real-time event availability tracking.

7. References

7.1 List of Sources

1. **Spring Boot Documentation**

   - "Spring Boot Reference Guide"
   - URL: https://docs.spring.io/spring-boot/docs/current/reference/html/

2. **Spring Security Documentation**

   - "Spring Security Reference Guide"
   - URL: https://docs.spring.io/spring-security/reference/

3. **MySQL Documentation**

   - "MySQL 8.0 Reference Manual"
   - URL: https://dev.mysql.com/doc/refman/8.0/en/

4. **Java Documentation**

   - "Java SE Documentation"
   - URL: https://docs.oracle.com/javase/8/docs/

5 **React Documentation**

   - "React Official Documentation"
   - URL: https://react.dev/