**SAVEETHA SCHOOL OF ENGINEERING**

**Saveetha Institute of Medical and Technical Sciences**

**Chennai – 602 105**

**BUS TICKETING AND PAYMENT SYSTEM**

**A CAPSTONE PROJECT REPORT**

***Submitted to***

**SIMATS ENGINEERING**

****

**By**

**P. Vishnu sai (192211432)**

***Supervisor***

**Dr. V. GOKULA KRISHNAN**

**CSA1584 – CLOUD COMPUTING AND BIG DATA ANALYTICS FOR MEDICAL APPLICATIONS**

**June – 2024**

**BUS TICKETING AND PAYMENT SYSTEM**

**Project Overview:**

The Bus Ticketing and Payment System is designed to streamline the process of purchasing and managing bus tickets for both passengers and operators. This system leverages modern technology to provide a seamless, efficient, and secure solution for ticketing and payment processes.

**1. Requirements Gathering:**

**Functional Requirements:**

* **User Authentication and Authorization:** Implement secure login and access control for passengers and administrators.
* **CRUD Operations for User Data:** Allow users to create, read, update, and delete their profiles and booking records.
* **Responsive User Interface:** Ensure the interface is accessible and functional across various devices (desktops, tablets, smartphones).
* **RESTful API for Data Access:** Provide a robust API to enable seamless data interactions between the frontend and backend.

**Non-Functional Requirements:**

* **High Availability and Scalability:** Ensure the system can handle high traffic loads and remain operational at all times.
* **Cost-Effectiveness:** Optimize resource usage to minimize operational costs.
* **Security and Compliance:** Implement strong security measures to protect user data and ensure compliance with relevant regulations (e.g., GDPR).
* **Performance and Load Balancing:** Maintain high performance under load and distribute traffic efficiently.

**2. Technology Stack:**

**Frontend:**

* **HTML, CSS, JavaScript:** Core technologies for structuring and styling the web application.
* **React.js:** A JavaScript library for building dynamic and responsive user interfaces.

**Backend:**

* **Node.js with Express.js Framework:** A runtime and framework for building a scalable and efficient server-side application.
* **RESTful API:** To enable communication between the frontend and backend services.

**Database:**

* **Amazon RDS (Relational Database Service) for MySQL:** Managed database service for storing and managing relational data.

**Infrastructure:**

* **Amazon EC2 (Elastic Compute Cloud):** Scalable compute capacity for hosting the backend services.
* **Amazon S3 (Simple Storage Service):** Storage for static files such as images and backups.
* **Amazon CloudFront:** Content Delivery Network (CDN) for low latency and high transfer speeds.
* **AWS Lambda:** Serverless computing for on-demand execution of certain tasks (optional).
* **Amazon Route 53:** Scalable Domain Name System (DNS) management.
* **AWS IAM (Identity and Access Management):** Secure access control and identity management.

**3. System Design:**

**Architecture Design:**

* **Frontend-Backend Communication:** Utilizing RESTful APIs for seamless data exchange.
* **Database Design:** Structuring the database to efficiently handle user data, ticketing information, and transaction records.
* **Scalability Considerations:** Designing the system to scale horizontally to handle increased load.

**User Interface Design:**

* **Wireframes and Prototypes:** Creating wireframes and prototypes for user interfaces to ensure a user-friendly design.
* **Responsive Design:** Ensuring the interface adapts to different screen sizes and devices.

**4. Development Phase:**

**Backend Development:**

* **Core Functionalities:** Developing user authentication, ticket booking, payment processing, and data management features.
* **API Development:** Creating robust RESTful APIs to enable frontend-backend communication.

**Frontend Development:**

* **User Interfaces:** Developing responsive and intuitive user interfaces for web and mobile applications.

**Integration:**

* **External Services:** Integrating with payment gateways and existing bus management systems.

**5. Testing Phase:**

**Unit Testing:**

* Testing individual components for functionality and reliability.

**System Testing:**

* Ensuring the entire system performs well and is secure.

**User Acceptance Testing (UAT):**

* Conducting testing with a group of end-users to gather feedback and ensure the system meets their needs.

**6. Deployment:**

**Deployment Plan:**

* Developing a detailed plan for deployment, including data migration and system setup.

**Pilot Launch:**

* Launching the system in a controlled environment to monitor performance and gather initial feedback.

**Full-Scale Deployment:**

* Rolling out the system to all users, ensuring minimal disruption to existing operations.

**7. Post-Deployment Support:**

**Monitoring and Maintenance:**

* Continuously monitoring the system for issues and performing regular maintenance.

**User Support:**

* Providing customer support to address user queries and issues.

**Updates and Enhancements:**

* Gathering feedback and implementing necessary updates and new features.

**8. Project Review:**

**Performance Evaluation:**

* Assessing the project's success against initial objectives and key performance indicators (KPIs).

**Lessons Learned:**

* Documenting lessons learned to improve future projects.

**Final Report:**

* Preparing a comprehensive report summarizing the project's outcomes, challenges, and achievements.