[G Santosh Mathavva] [23026952]

[W64L] | [DIT]

Software Application Development

CA1: Web Application development proposal

|  |
| --- |
| 1. **App Ideas**   **Concert Event Ticket Buying Platform**  The Concert Event Ticket Buying Platform is a web application designed to streamline the process of discovering, purchasing, and managing tickets for concerts and live music events. The platform will serve both concert organizers and music fans by offering tools for event creation, ticket sales, seat selection, and event notifications. The potential impact includes simplifying the ticket buying process, enhancing user experience, increasing event attendance, and providing organizers with comprehensive event management tools. |
| 1. **Key Features and Functionalities**  * **User Authentication:** * **Sign Up/Login:** Users can create an account or log in using their credentials. * **Role Management:** Different roles (admin, organizer, attendee) with specific permissions. * **Event Management:** * **Create Event:** Organizers can create new concert events with details such as name, date, time, location, description, and ticket pricing. * **Retrieve Events:** Users can view a list of upcoming concerts and access detailed event pages. * **Update Event:** Organizers can modify event details and update information as needed. * **Delete Event:** Organizers can delete events that are canceled or no longer relevant. * **Ticket Management:** * **Purchase Tickets:** Users can buy tickets for events, select seats (if applicable), and complete payment through an integrated payment gateway. * **Retrieve Tickets:** Users can view their purchased tickets and download them. * **Update Ticket Information:** Users can update their personal information associated with their tickets. * **Cancel Tickets:** Users can cancel their tickets if the event supports cancellations and refunds. * **Event Notifications:** * **Send Notifications:** Organizers can send email notifications to attendees about event updates, reminders, and promotions. * **Receive Notifications:** Attendees can opt-in to receive notifications about upcoming events and ticket availability. * **Search and Filter:** * **Search Events:** Users can search for events by artist, venue, date, or genre. * **Filter Events:** Users can filter events based on criteria such as location, date range, and price range. |
| 1. **Database Fields**  |  |  |  | | --- | --- | --- | | **Field** | **Data type** | **Special** | | **User Table** | | | | user\_id | int, primary key |  | | user\_name | varchar, unique |  | | password | varchar |  | | email | varchar, unique |  | | role | varchar |  | | **Events Table** | | | | event\_id | int, primary key |  | | organizer\_id | int, foreign key |  | | event\_name | varchar |  | | event\_date | date |  | | event\_time | time |  | | event\_location | varchar |  | | event\_description | text |  | | ticket\_price | decimal |  | | **Tickets Table** | | | | ticket\_id | int, primary key |  | | event\_id | int, foreign key |  | | user\_id | int, foreign key |  | | seat\_no | varchar, nullable |  | | purchase\_date | date |  | | status | varchar |  | | **Notifications Table** | | | | notification\_id | int, primary key |  | | event\_id | int, foreign key |  | | user\_id | int, foreign key |  | | message | text |  | | sent\_date | date |  | |
| 1. **Target Audience**  * **Concert Organizers**: Individuals or companies organizing live music events. * **Music Fans:** People of all ages who enjoy attending live music concerts. * **Artists and Bands**: Musicians looking to promote their events and sell tickets. * **General Public:** Anyone interested in exploring and attending music events. |
| 1. **Potential Challenges**  * **Scalability:** Ensuring the platform can handle high traffic during peak ticket sale times. * **Data Security:** Protecting user data and payment information from breaches. * **Fraud Prevention:** Preventing ticket fraud and unauthorized transactions. * **Performance:** Maintaining fast response times and smooth user experience. |
| 1. **Timeline for Development**  * **Planning Phase (2 weeks):** * Requirement gathering Define project scope Design database schema * **Design Phase (3 weeks):** * Create wireframes and UI designs * Design RESTful API endpoints * **Development Phase (8 weeks):** * Set up project structure and environment * Develop user authentication and role management * Implement CRUD operations for events and tickets * Develop ticket purchasing and notification features * Conduct unit and integration testing   **Milestones:**  **Week 1-2:** Planning completed  **Week 3-5:** Design finalized  **Week 6-13:** Development completed |
| 1. **Additional Features for Future Iterations**  * **Loyalty Program:** Implement a loyalty program to reward frequent attendees * **Virtual Concerts:** Support virtual concerts and live streaming options for remote attendees. * **Secure Login:** Users can securely login to their accounts by enabling * 2 Factor Authentication(2FA) and/or Facial recognition. |