Project Management Document for Concert Ticket Site

Definition of the Tasks Involved in the Project

The concert ticket-buying platform aims to provide an accessible, efficient, and secure system for purchasing concert tickets. The platform will cater to both concertgoers and event organizers.

Tasks include:

- **Market Research**: Conduct questionnaires and competitor analysis.
- **Backend Development**: Building server-side logic to manage user accounts, ticket inventory, and transactions.
- **Frontend Development**: Designing and implementing the user interface for intuitive navigation.
- **Accessibility**: Ensuring compatibility with screen readers and mobile browsers.
- **Testing and Usability**: Running user studies to evaluate the platform's ease of use.

Risk

Potential risks to the company include:

- **Monetary Risk**: High initial setup and maintenance costs for servers and payment processing systems. To mitigate this, we will employ cloud hosting and scalable solutions like AWS or Azure.
- **Scalability Issues**: As the platform grows, we might face challenges managing high traffic during popular ticket sales. We will address this by using load balancing and scalable database solutions.

Potential risks to users include:

- **Data Breach**: Sensitive information, such as payment details and personal data, could be at risk. To mitigate this, we will implement strong encryption for all transactions and use secure third-party payment processors.
- **Scalping**: Scalpers may try to buy large numbers of tickets for resale. Our verified fan system will reduce this risk by giving early access to real fans and limiting bulk purchases.

Evaluation Criteria and Methods

We plan to run a user study to evaluate usability. Feedback will be gathered on ease of ticket purchasing, navigation, and overall satisfaction.

Implementation

General tasks include:

- **HTML, CSS, and JavaScript**: Frontend development for responsive design.
- **Node.js, Express.js**: Backend logic for handling user sessions, events, ticket inventories, etc.
- **Database Management**: Implementing an SQL database for storing user, event, and ticket data.
- **Version Control**: Using Git and GitHub for code collaboration.

Specific features to be implemented can be found in the requirements document.

Training

The platform will feature a comprehensive **FAQ page**, as well as a **Contact Us** form for customer support inquiries. The support "team" will handle customer requests and provide assistance for any technical difficulties.

Maintenance

Post-deployment, maintenance tasks will include:

- **Bug Fixes**: Addressing user-reported issues.
- **Security Patches**: Regularly updating the platform to mitigate security risks.
- **Performance Monitoring**: Ensuring the system runs efficiently with increased traffic.
- **Content Updates**: Adding new concerts, merchandise, and event organizer tools.

Future Needs

In future versions, we plan to introduce features like advanced payment processing (support for multiple payment gateways), artist and organizer ID verification, and a mobile app for easier access. Additionally, we envision automated updating for event listings, merchandise integration, and further measures to combat ticket scalping using bot detection tools.