Event Planner

Requirements and Context:

The requirements are to develop an event platform that serves as an all-in-one dashboard, integrating upcoming events, invites, suggestions, and social interaction.

Context includes the need for a user-friendly interface that caters to both virtual and physical events, with a focus on essential event information and smart event suggestions.

Stakeholders/Target Users:

Event organizers, attendees, and possibly sponsors.

Individuals interested in finding, attending, and organizing events.

Content:

The content of the platform should include event details, social interactions, feedback and reviews, and event materials (tickets, maps, brochures, announcements).

Technologies to Use:

Web development technologies for a responsive and accessible user interface.

Frontend Technologies:

HTML5 and CSS: For structuring and styling the user interface.

JavaScript (ES6+): For client-side scripting and enhancing user interactivity.

React or Vue.js: Modern JavaScript libraries/frameworks for building interactive and responsive user interfaces.

Backend Technologies:

Node.js: A server-side JavaScript runtime for building scalable and efficient backend applications.

Express.js: A web application framework for Node.js that simplifies backend development.

MongoDB or PostgreSQL: Databases to store user data, event details, and other relevant information.

AI for smart event suggestions.

Social media APIs for integration.

Database for storing user preferences and event details.

Design Questionnaire:

Questions:

Where do users find events?

What social platforms do users use for event-related discussions?

What features do users consider essential in an event dashboard?

How do users prefer to receive event invitations?

Possible Means of Data Gathering:

Online Surveys:

Distribute surveys to a broader audience to collect quantitative data on preferences and expectations.

Next Steps:

Prototyping: Begin creating low-fidelity prototypes based on the wireframes.

User Testing: Conduct usability testing with potential users to gather feedback on the prototype.

Iteration: Refine the prototype based on user feedback.

High-Fidelity Prototypes: Develop high-fidelity prototypes incorporating final design elements.

Development: Move forward with full-scale development based on the approved prototypes.