Catchphrase Web App

Step 1. Project Planning and Specifications

Purpose: To develop a Catchphrase Web App that allows users to create and join game rooms, play a card-based game, and keep track of scores.

Specifications:

* Users can create a game room and get a unique game code.
* Users can join existing game rooms using a game code
* In the game room, users can set their names and teams
* The party leader can then start the game
* The game randomly selects a player to start, and words/phrases from the category for that player to describe
* Players take turns describing the word/phrase to their team without using it
* Teams guess word/phrase within a time limit
* Points are awarded for correct guesses
* Players are able to skip words/phrases to obtain a new one.
* Team with most points wins the game

Step 2: User Stories

* As a player, I want to create a game room with unique codes
* As a player, I want to be able to join game rooms using codes
* As a player, I want to set my name and team in the game room.
* As a player, I want to describe words/phrases to my team
* As a player, I want to see the remaining time during everyones turn
* As a player, I want to see the score and status
* As a player I want to see the winning/losing team at the end of the game

Step 3: Database Design(MongoDB)

Collections:

Users: Player names

GameRooms: To store game room information – code, players, teams, categories,

Categories: To store words/phrases

GameHistory: To store scores

Step 4: Tech Stack

Frontend :React with Typescript, Redux for statemanagement, Material-UI for UI components

Backend: Node.js with Express, Typescript, and MongoDB for database

Real-Time Communicatoin: Socket.io

Step 5: Routes and API’s

* Api/game : Game room creation, joining and management.
* Api/categories: Fethcing and managing categories
* Api/gameplay : Handling game logic (start game, rounds, scoring ect.)

Step 6: Story boarding

Create wireframes/story boards for the UI. Including Front page, create game, join game, select category, gameplay, and end-game results.

Step 7: Front End Development

1. Create components for game room creation/joining and team selection
2. Develop a category selection component.
3. Implement game logic for rounds with a timer.
4. Create a scoreboard component to display scores.
5. Develop end-game screens for winners and losers

Step 8: Backend Development

1. Implement game creation/joining logic.
2. Api for category management (fetching, creation, voting, selection)
3. Implement game logic (game start, rounds, scoring)
4. Implement WebSocket for real-time updates during the game.

Step 9: Testing

Perform unit testing and integration testing for both front/backend components. Test various scenarios, including creating/joining game rooms, gameplay, and scoring.

Step 10: Deployment

Deploy the front and backend to hosting service. Set up production ready MongoDB

Step 11: User Acceptance Testing

Invite users to test application and provide feedback. Address any issues or bugs identified

Step 12:Documentation

Documentation on how to use the app, including game rules and controls.

Step 13: Maintenances/Updates

Update application, address bug fixes, adding new features based on user feedback.