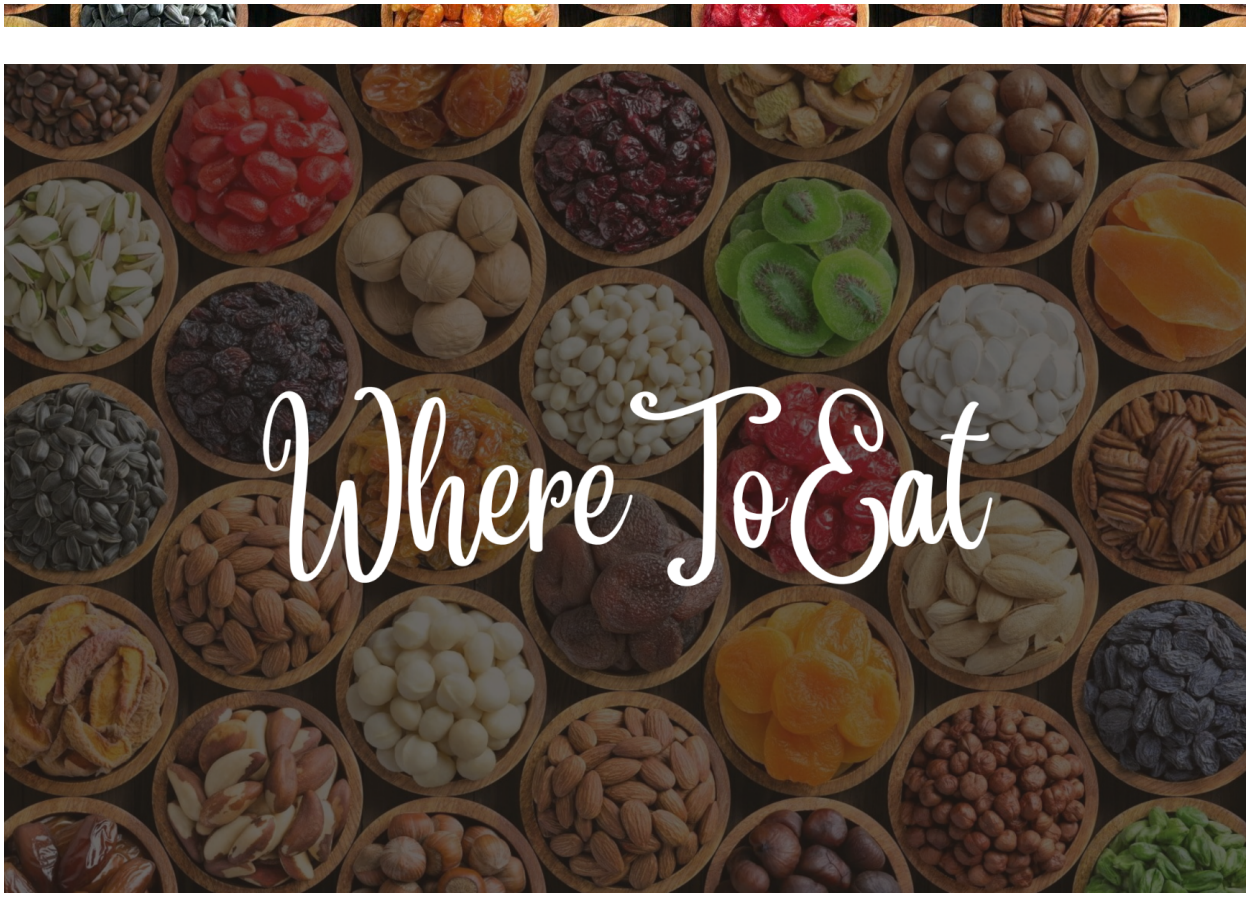


WhereToEat

Team Noobs

Ibrahim Alkuwaifi

Sai Kumar Gurram

Teja Peddi

Ashesh Kumar Pinisetti

Mohammad Khaja Moinuddin

Vishal Kandunuri

Mohammed Abdul Wahid

Gouthami Gongidi

Mallikarjun Narra

CSCE 54320.008

Description

The "WhereToEat" app is a social dining decision-making platform designed to simplify the process of choosing a restaurant when dining out with friends or family. This web application will provide a user-friendly interface for creating, joining, and interacting with hangouts, allowing users to collectively decide on a restaurant while incorporating various features for a seamless and enjoyable experience.

Development Environment

1. Front-end Framework: Angular
2. Back-end: Firebase (Authentication and Firestore)
3. Third-party API: Google Places API (for restaurant data)
4. UI/UX Design: Bootstrap and Angular Material
5. Deployment: Firebase Hosting

Key Features

I. Authentication

Users can sign up or log in using Firebase Authentication, ensuring secure access to the app. Only authenticated users can create and join hangouts. Users can also reset their password if they forgot it.

II. Hangout Creation

Hangout creators can specify the hangout's name, location, and an optional description. They can also choose whether users can add their own restaurants or generate random restaurants.

III. Adding Restaurants

Users can add their own restaurants, they can provide restaurant details, including name, address, cuisine type, and other relevant information. They can also manually add nearby restaurants using the Google Places API. If generating random restaurants, the app uses the Google Places API to fetch a list of nearby restaurants.

IV. Restaurant Voting

All hangout participants can view and vote for their preferred restaurant. Real-time vote calculations are displayed, and the winning restaurant is highlighted.

V. Hangout Interaction

Users can join hangouts created by others, view hangout details, and see the list of restaurants and their voting status. An intuitive interface allows users to vote and change their votes as needed.

VI. Data Management

Firestore is used to store hangout details, user information, and restaurant data. Firestore collections are structured for hangouts, users, and restaurants, ensuring efficient data management.

VII. User Profile

Users can view and update their profiles, displaying user-specific information and the hangouts they've joined or created.

Project Timeline:

Project Duration: Fall 2023 (Start Date: August 21, 2023 - End Date: December 15, 2023) approx.

Note: All the dates are subject to change.

Tasks/ Milestones:

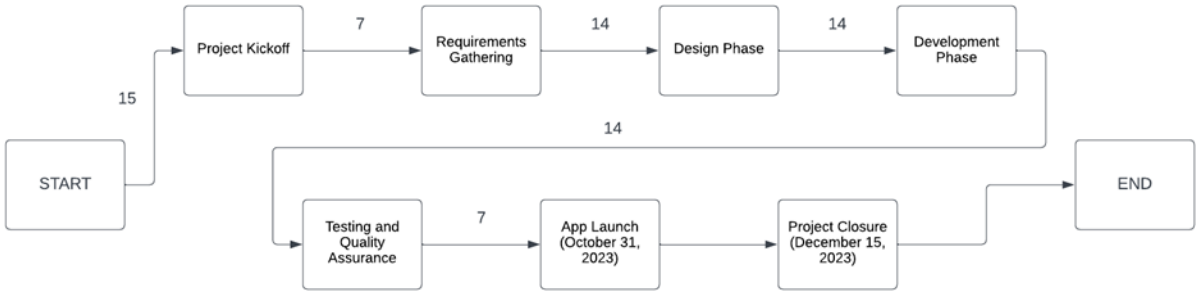
- Project Kickoff - (08/21/2023)
- Requirements Gathering -Team - (09/05/2023)
- Design and Planning Phase - (09/11/2023)
- Development Phase - (10/02/2023)
- Testing and Quality Assurance - (10/23/2023)
- App Launch - (11/13/2023)
- Project Closure - (12/4/2023)

Dependencies:

- The older tasks must be completed before the new task can be created.
- The mockups must be completed before the code can be developed.
- The code must be completed before the website can be tested.

Charts:

PERT Chart



Gantt Chart:

Note: The Gantt chart is attached to a Google sheet. Please click here: https://docs.google.com/spreadsheets/d/1S7VeOyoBf7qaHQSynllYPHKLLXgi3WY1i7qQzK-065Q/edit?usp=sharing

Risk Management

Security and Privacy:

Monitoring:

- Make sure user data is protected properly and monitor firebase security updates or vulnerabilities(Just in case any change might affect our users).
- Implement logs to track failures or unauthorized access.

Re-evaluation:

- By following security best practices.
- Review and update security protocols(if any) based on security updates and user feedback.

Contingency Plan:

- In case of a security incident, we will notify affected users, while investigating the breach.

Dependency on Third-Party Services:

Monitoring:

- Frequently monitor the status and availability of third-party services, especially Firebase and the Google Places API.

Re-evaluation:

- By regularly assessing the impact of third-party service changes on our app's functionality

Contingency Plan:

- If a critical third-party service experiences downtime, we will make sure the users are notified of our application downtime(to limit the scope of this project).

Codebase Complexity and Maintainability:

Monitoring:

- Keep track of technical debt and code quality issues or bugs that are identified during code reviews.

Re-evaluation

- Periodically review the codebase for areas that have become complex or difficult to maintain.
- We will conduct code refactoring or rewriting of critical components if code quality issues persist.

Contingency Plan:

- If the codebase becomes complex and difficult to maintain, allocate time and resources for refactoring efforts to simplify and improve code quality.
- Establish coding standards and guidelines for the development team to ensure consistency of the codebase.

Team Members Roles

Project Management Lead (Ibrahim Alkuwaifi):

- Responsible for overall project planning, coordination, and ensuring that the project stays on track.
- Manages timelines, deadlines, and project milestones.
- Facilitates communication between team members.
- Ensures the project meets its objectives and is delivered successfully.

Requirements Lead (Gouthami Gongidi):

- Gathers and documents project requirements, ensuring they align with the app's objectives.
- Collaborates with stakeholders to define user stories and feature specifications.
- Ensures that the development team understands and implements the specified requirements.

Design Lead (Ashesh Kumar Pinisetti):

- Leads the design phase of the project.
- Creates wireframes, prototypes, and designs the user interface.
- Works closely with the UI/UX team to ensure a visually appealing and user-friendly design.

Implementation Lead for Front-end (Gurram Sai Kumar):

- Leads the front-end development efforts using the Angular framework.
- Translates the design into a functional user interface.
- Collaborates with the back-end team for seamless integration.

Implementation Lead for Back-end (Mallikarjun Narra):

- Leads the development of the back-end using Firebase.
- Implements user authentication, database management, and server logic.
- Collaborates with the front-end team to ensure data integration.

Configuration Management Lead:

- Manages version control and configuration management for the project.
- Ensures that codebase changes are tracked and documented effectively.
- Helps maintain code quality and consistency.

Testing Lead (Vishal Kandunuri):

- Manages the testing phase of the project.

- Creates test plans, conducts testing, and reports and tracks issues.
- Ensures the application meets quality standards and is bug-free.

Documentation Lead (Teja Peddi):

- Creates and maintains project documentation.
- Develops user guides, technical documentation, and FAQs.
- Ensures that project documentation is up-to-date.

Demo and Presentation Lead (Wahid Mohammed):

- Prepares and delivers project demos and presentations.
- Showcases the app's features and functionality to stakeholders and potential users.
- Helps create a compelling narrative for the project.

System Administrator Lead (Mohammad Khaja Moinuddin):

- Manages the system administration tasks, including deployment and server maintenance.
- Ensures that the web application is hosted and accessible via Firebase Hosting.
- Monitors server performance and addresses any technical issues.