EECS 581 Project 3

Team Number
Team 12

Team Members
- Liam Aga
- Aiden Barnard
- Reem Fatima
- Jorge Trujillo
- Alejandro Sandoval
- Pashia Vang

Project Name
Side Quest

Project Synopsis
A website that gives fun campus challenges and tracks your progress.

Architecture

Overview
Side Quest is a website / web app that makes exploring the University of Kansas campus more fun.
Instead of focusing only on halls, the website covers *any location on campus* - such as buildings, landmarks, and secret spots.
At each location, users can see short, fun challenges like:
- “Ride the sketchy elevator in Summerfield.”
- “Find the Jayhawk hidden near Strong Hall.”
- “Get a coffee from The Underground.”

When a user completes a challenge, they can check it off. The app keeps track of everything through a personal account, so progress stays saved across devices. This also allows progress to be compared between all users with statistics shown in the website.

Core Features
1. **Campus Locations**
    - Includes all types of spots: halls, libraries, and outdoor areas.
    - Each location has its own quests and short descriptions.
2. **Quest List**

    - Shows fun tasks or dares you can do at each place.
3. **User Accounts**
    - Each user can sign up or log in.
    - Tracks completed quests and total progress.
4. **Progress Tracking**
    - Displays how many quests have been completed and which locations are still open.
    - Saves progress automatically under each user's account.
5. **Checkmark System**
    - Users can mark a quest as completed with a tap.
6. **Community Interaction**
    - Users can see their progress in leaderboards and compete for quest completion.
7. **Location Additions (Future Addition)**
    - Allow administrators to provide location and quest updates without modifying the code.

User Flow

1. The user opens the web app and logs into their account.
2. They see a map or list of campus locations.
3. They pick a location (for example, *Watson Library*).
4. The web app shows quests for that spot.
5. The user completes a quest and taps the checkmark.
6. Progress updates in their account automatically.

System Design

1. Diagram – High-Level Overview

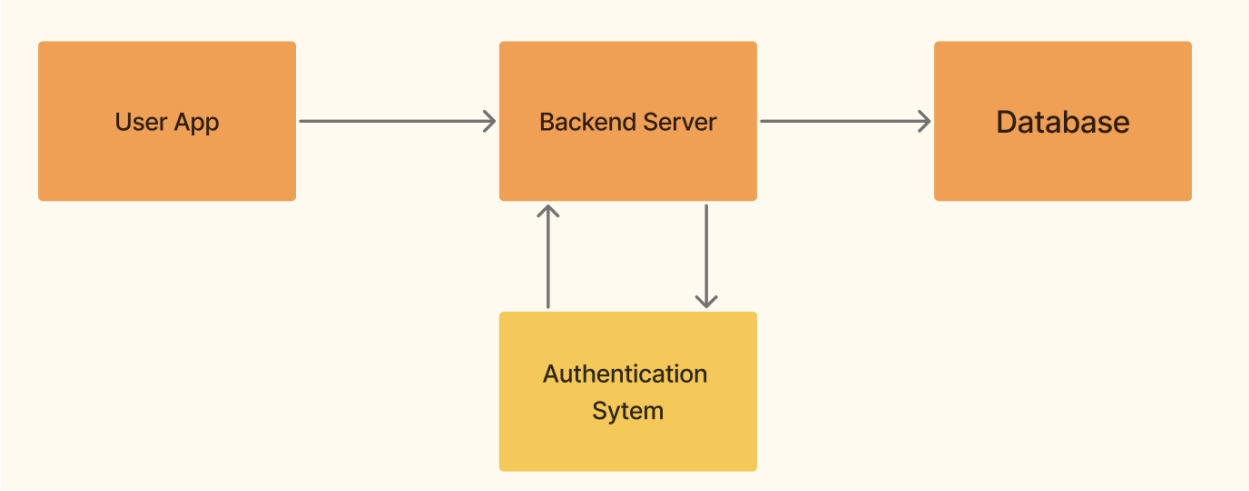

Figure 1: High-Level System Overview

- The **User App** displays data and lets users interact with quests.
- The **Backend Server** handles requests, sends data, and updates progress.

- The **Authentication System** manages user accounts and login sessions.
- The **Database** stores users, locations, and quest progress.

2. Diagram - Data Flow

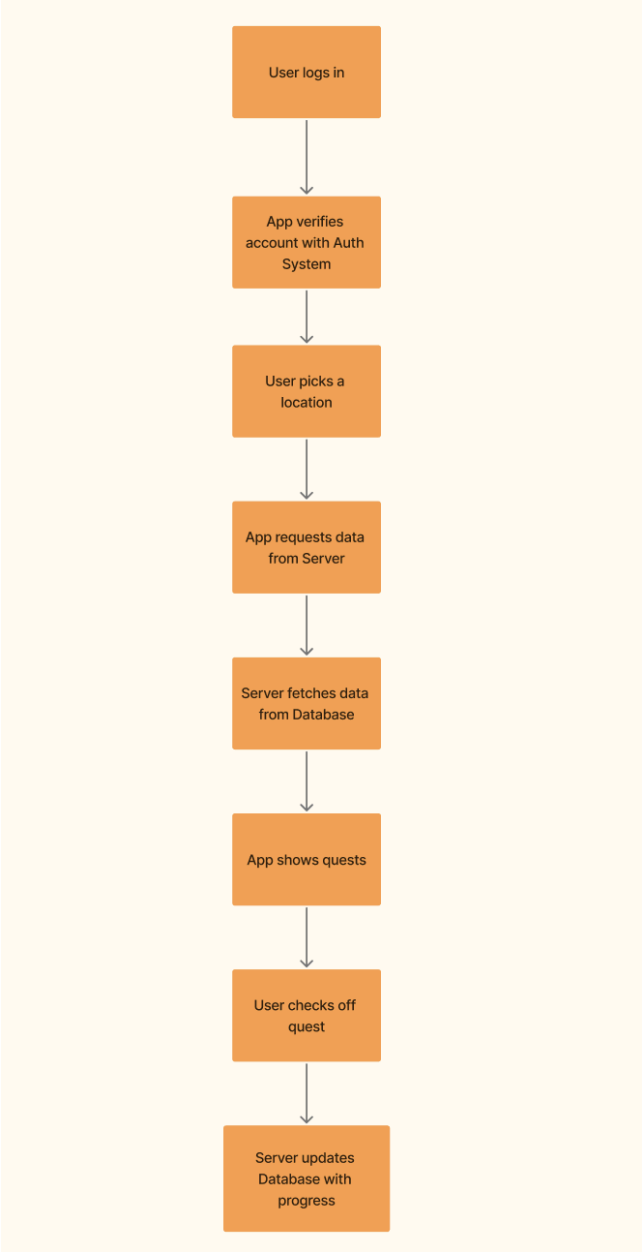

Figure 2: Data Flow Overview

3. Diagram - Example Data Model

A database will be used to store user account information.

Table	Fields	Description

Users	user_id, name, email, password_hash	Stores login and profile info
Locations	location_id, name, type, description	Info about any KU location
Quests	quest_id, location_id, text	List of challenges for each spot
Progress	progress_id, user_id, quest_id, completed	Tracks which quests a user finished

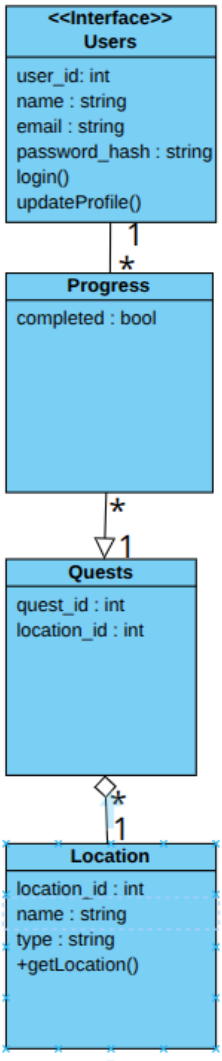

Figure 3: Example Data Model

Data Flow Example

1. The user first logs in with their Side Quest account.

2. When they view a location, the app fetches that location's quests.
3. After completing one, the user marks it done.
4. The app updates the *Progress* table under their account.
5. Their personal dashboard shows total quests completed and unlocked spots.

Technologies

- **Frontend & Backend:**
  The app uses Next.js for both the interface and the backend. It keeps everything in one place so it's easy to build pages, connect routes, and handle data. This setup also makes the app run fast and stay organized as more features are added.

- **Database:**
  All data is stored in Supabase, which uses PostgreSQL. It holds information like user accounts, locations, and quest progress. The database updates quickly when users make changes and keeps everything structured so the app runs smoothly.

- **Authentication:**
  Supabase Auth manages user sign-up, login, and sessions. It keeps accounts safe and remembers users when they come back. This makes it simple for players to log in and continue their progress without any extra steps.

User Experience Goals

- Clean and colorful design that feels fun and welcoming.
- Checkmarks and progress bars to show completion clearly.
- Easy navigation - users should find new locations and quests fast.
- Lighthearted tone with short, funny quests.
- Easily accessible, the web app should be compatible on various devices.

Future Plans

- Add a campus map that highlights completed and uncompleted locations.
- Introduce a **leaderboard** for most active users.
- Allow users to share their achievements with friends.