Project Proposals

1. Weather application
   1. Tracks weather based on users’ location
   2. User Table
   3. Locations associated with the user
   4. Grab weather data from 3rd party API
2. Auto shop application
   1. Track orders for an auto shop
   2. Table for brands of cars
   3. Car parts table
   4. Customer table
3. Company Management System
   1. Organize event tasks within a company
   2. Employee table
   3. Manager table
   4. Table for events and tasks
4. Baseball hitter game
   1. It is more complex but hits every requirement and more
   2. Database for users, scoreboard, equipment, fields so on
   3. The user signs up. To play the game a user predicts where the ball will be, and depending on where the ball gets randomly put, the user receives either a strike or a hit, and depending on how good the guess was will determine how good the hit was.
5. Music player app/ movie app either or
   1. Database for users, music, and artist
   2. Use crud operations for the songs, add a few user stories on suggestions to other songs, and another story for a user getting all music made by one artist.
   3. Find similar music that other users listen to such as genre and artist

### **Weather Application**

**Description**: A weather application that provides users with real-time weather information, forecasts, and weather alerts. Users can view weather details for multiple locations and receive notifications for severe weather conditions.

#### **User Roles**

* **Guest**: Can view basic weather information for a default location.
* **Registered User**: Can save multiple locations, set preferences, and receive notifications.
* **Admin**: Manages user accounts and configures application settings.

#### **Features**

* **Weather Information**:
  + Current weather conditions (temperature, humidity, wind speed, etc.).
  + 7-day weather forecast.
  + Hourly weather forecast.
  + Weather alerts and warnings.
* **User Management**:
  + User registration and login.
  + Profile management (update preferences, change password).
  + Location management (add, remove, and set primary location).
* **Notifications**:
  + Severe weather alerts.
  + Daily weather summary.
* **Integration**:
  + Integration with third-party weather APIs (e.g., OpenWeatherMap, Weatherstack).
* **Security**:
  + JWT-based authentication for registered users and admins.
* **Stretch Goals**:
  + Weather radar and map.
  + Integration with social media for sharing weather updates.

### **Automobile Parts Application**

**Description**: An e-commerce application for buying and selling automobile parts. Users can search for parts, view details, and place orders. Sellers can list parts and manage inventory.

#### **User Roles**

* **Customer**: Can browse parts, add items to cart, and place orders.
* **Seller**: Can list parts, manage inventory, and view orders.
* **Admin**: Manages users, reviews listings, and handles order disputes.

#### **Features**

* **Product Management**:
  + CRUD operations for parts (name, description, price, images).
  + Inventory management (stock levels, out-of-stock notifications).
* **Order Management**:
  + Shopping cart functionality.
  + Order placement and tracking.
  + Order history and status updates.
* **User Management**:
  + User registration and login.
  + Profile management (update personal information, change password).
  + Seller account management.
* **Search and Filtering**:
  + Advanced search (by part number, name, category).
  + Filtering by price, brand, and compatibility.
* **Payment Integration**:
  + Integration with third-party payment gateways (e.g., Stripe, PayPal).
* **Notifications**:
  + Order confirmations and updates.
  + Stock level alerts for sellers.
* **Security**:
  + JWT-based authentication for customers, sellers, and admins.
* **Stretch Goals**:
  + Reviews and ratings for parts and sellers.
  + Integration with shipping APIs for real-time shipping quotes.

### **Event Management System**

**Description**: A platform for organizing and managing events. Users can create events, invite attendees, and manage event details. Attendees can register for events and receive updates.

#### **User Roles**

* **Event Organizer**: Can create and manage events, view attendee lists, and send updates.
* **Attendee**: Can browse events, register for events, and receive notifications.
* **Admin**: Manages users and events, reviews reports, and handles disputes.

#### **Features**

* **Event Management**:
  + CRUD operations for events (name, description, date, location, images).
  + Ticketing and registration management.
  + Event schedule and agenda management.
* **User Management**:
  + User registration and login.
  + Profile management (update personal information, change password).
* **Attendee Management**:
  + Event registration and RSVP.
  + Attendee lists and contact information.
* **Notifications**:
  + Event reminders and updates.
  + Registration confirmations and cancellations.
* **Integration**:
  + Integration with third-party calendar APIs (e.g., Google Calendar).
  + Payment gateways for event ticketing.
* **Security**:
  + JWT-based authentication for event organizers, attendees, and admins.
* **Stretch Goals**:
  + Event feedback and rating system.
  + Social media integration for event sharing.
  + Analytics and reporting for event performance.

Outline tables and modules going to be using