Final Project Summary

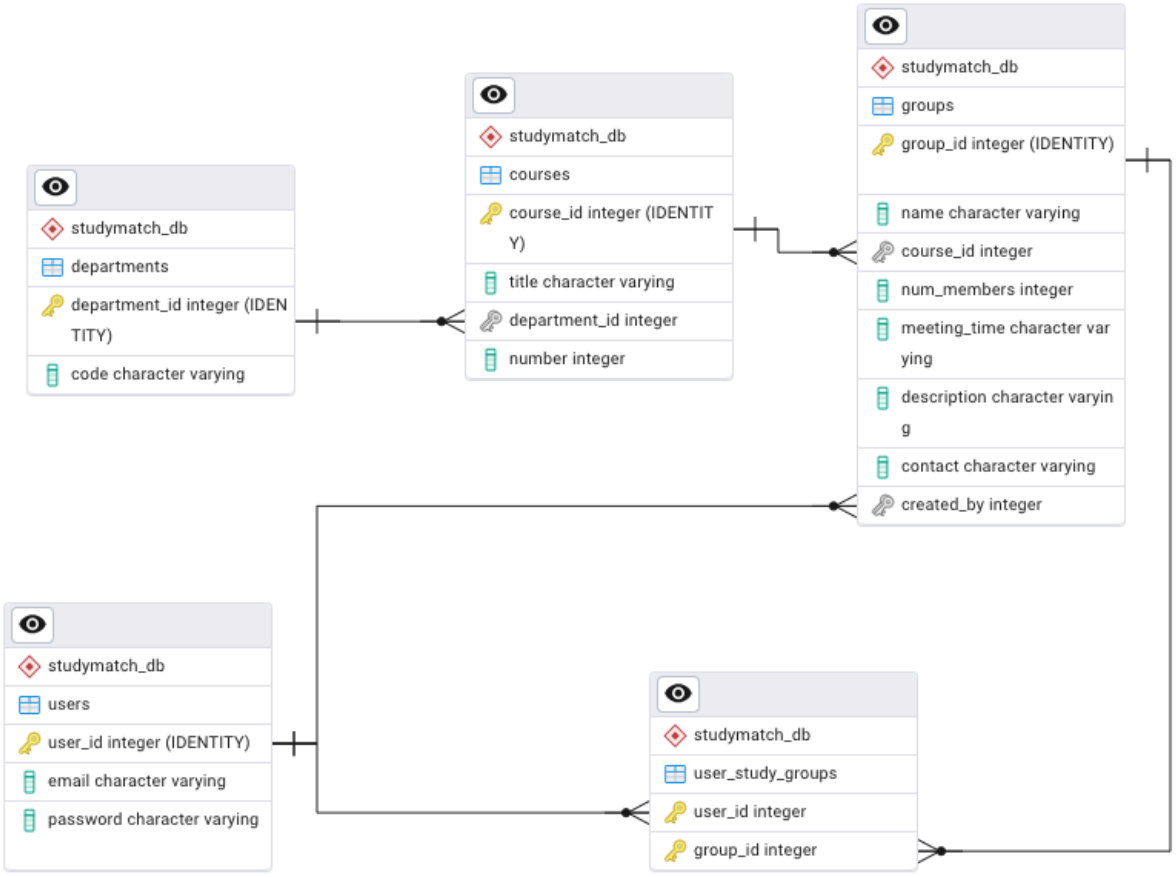
StudyMatch

Topic/Purpose:
StudyMatch is a full-stack web application designed to help students at USC find and join study groups within their classes. The goal is to foster academic collaboration by making it easy for students to discover groups in their courses.

How the Site Works:
Users can register and log in. When not logged in users can look at courses and view study groups within each course. Once logged in, students can: join study groups and create a new group. Once they join a group they can see the contact email of the group so they can contact about how the group meets up. If they are the creator of the group they can edit or delete the group info.

Source of data:
Course data was preloaded manually into the database from USC web registration. Group and user data are dynamically created and updated when users interact with the site.

Database Diagram:



Features Implemented:

1. Form:
StudyMatch includes several forms that serve meaningful functions:

Login and Signup Forms to authenticate users.

The Create Group Form allows users to input a name, course, meeting time, and description to generate a new study group.

The Edit Group Form lets users update group details.

2. Event-Driven DOM Manipulation:
The interface dynamically responds to user actions:

When users join or leave a group, the page updates the group card like the number of members, contact email, and adding/removing groups from the user's group list without needing a full reload.

3. User Input Validation:
User input is validated on the client side:

Forms prevent submission with empty required fields or invalid input.

Users see error messages for invalid credentials, missing required fields, and failed login/register attempts.

4. Authentication:
StudyMatch implements a session-based authentication using express-session and connect-pg-simple:

User credentials are stored in the database.

Sessions are tracked using cookies.

Authenticated users gain access to login-protected pages like group creation and joining.

5. REST API Back-End with React Front-End:

React Front-End: Manages the user interface, routing, and handles user interactions.

Express REST API Back-End: API endpoints are defined which the frontend calls to perform database operations like creating, reading, updating, and deleting data (CRUD). Communication happens via fetch() requests, with JSON responses passed between client and server.

7. Many-to-Many Table Relationship:

User_study_groups table links users and groups.

This allows each user to be in multiple groups and each group to contain multiple users.

CSS Frameworks / Templates
The site uses custom CSS and Bootstrap for components and layout.