Documentation

Outline

Big Boulder is a fitness progress tracking application designed for bouldering enthusiasts. The app features a user registration and login, ability to log the climbs you recently have done, view your previous climbing sessions, and even converse with other climbers on the forum. It's built using Node.js, Express, EJS and MySQL. User data and logged climbs as well as forum posts are all stored in the MySQL database. The app is deployed on the Goldsmiths virtual servers and installable locally from the GitHub repository.

Architecture

The application uses a 2-tier system, Front End and Back End.

Front End:

> Application tier: Node.js + Express for routing, middleware, session handling, authentication and rendering EJS templates.

Back End:

> Data tier: MySQL database for storing persistent data such as users, climb data, forum posts.

Data Model

Tables:

- **Users –** stores account information (user ID, username, email, password, date of account creation)
- **Climbs –** stores climb information (climb ID, ID of the user it belongs to, name given, difficulty, date added)
- **Posts –** stores post information (post ID, ID of the user posting, title, post content, date created, and ID of the parent post)
- **Audit –** stores login audits (date of event, username, whether it was successful, event type)

User Functionality

- Home page – a home page that briefly goes over what the app is for and buttons to link the user to the login/registration page when they are not logged in, or add climb and sessions page if they are.
- About Page – A page that contains more information on the app. I wrote a mockup about page, since this app obviously doesn't really exist.
- Authentication – The user can click "Login" to go to the login page and alternatively register a new account. The default login details are username:

"gold", password: "smiths123ABC$". Password validation matches coursework requirements and is set to a minimum of 8 characters.

- Climbs – A logged in user can add climbs they have been doing at their climbing gym, they can enter a name they would like to give the climb, and enter the difficulty. Validation requires both fields.
- Sessions – The My Sessions page allows the user to see their past sessions and climbs. Grouped by session date and lists all climbs during that day.
- Forum – The forum allows both logged-in and not-logged-in users to read posts made by other users. Logged-in users can also create posts, and reply to posts on the forum.

Advanced Techniques

1. Recursive Template Rendering
My *forum.ejs* uses a recursive *renderPost()* function to handle nested replies at arbitrary depth - this is server-side templating that builds a hierarchical view from a flat database.

2. Self-Referential Database Schema
The posts table with *parentpost* foreign key pointing to itself enables infinite threading depth.

3. Dynamic UI Components
Auto-expanding textareas with client-side JavaScript that responds to user input in real-time and animated buttons.

AI Declaration

I used AI for troubleshooting and finding and fixing bugs. I also used AI to write the content of the home and about pages to save time. I also used AI to help me write a readme to save time. All the final code is my own and written by me.