

Study Buddy

Technical Design

Team Heart of Gold

Database Components

- package.json
 - This JSON file will contain the necessary information about our application, as well as keep track of our dependencies:
 - Express
 - mySQL

Database Components

- user.sql
 - This mySQL database file will create a table for a user which includes:
 - id, name, age, profile picture, active, bio, school, courses, study preferences, chat history, matches, pending buddies, and buddies

Database Components

- user.js
 - This javascript file will export the following functions:
 - find -- returns a specific user from the database
 - update -- allows for existing entries to be updated
 - insert -- allows for new users to be inserted in db

Data Translation Components

- index.js
 - A router which will provide GET methods for
 - the home page view
 - login
 - profile creation (possibly ajax)
 - password recovery
 - And POST methods for
 - authentication
 - processing information for user creation

Data Translation Components

- users.js
 - A router which will (after users have signed in) provide GET methods for
 - the main user view (home)
 - logout
 - user editing/deletion
 - matching
 - chat

Frontend/UI Components

- `studystyle.css`
 - This will be a universal stylesheet for the website that will create a universal look-and-feel
 - This is a base-level component and has no dependencies

Frontend/UI Components

- `doj.js`
 - This file will have scripting that is used for a wide range of functionality on the website, and there may be other `.js` files that are made for the login and registration screens.
 - This is a base-level component and has no dependencies.

Frontend/UI Components

- index.html
 - This HTML document will be the first thing a user sees when they visit the application, and it will have a login option, with an included integration for password recovery and creating a new login registration if necessary.
 - Requires:
 - studystyle.css, doj.js
 - Backend management and data parsing

Frontend/UI Components

- `profBuilder.html`
 - This document will basically be the long-form version of being able to build a profile for a user. The data will be submitted to the database by form after the user completes the document.
 - Requires:
 - `studystyle.css`, `doj.js`
 - Backend management, data translation including Ajax protocols

Frontend/UI Components

- home.html
 - This document will have a list of new matches, old matches, classes that are being taken, etc.
 - Requires
 - studystyle.css, doj.js
 - Backend management and data parsing

Frontend/UI Components

- `matcher.html`
 - This will be the actual “matching” portion of the application that will present matches based on attributes the user has added in their profile.
 - Requires:
 - `studystyle.css`, `doj.js`
 - Backend management and data parsing

Frontend/UI Components

- chat.html
 - This will be a dynamic view that will allow someone to chat with their matches and set up meetings to study with them. It will allow you to choose a match, which will open a chat window. The user will only be able to chat with one other user at a time.
 - Requires:
 - studystyle.css, doj.js
 - Backend management and data parsing

Frontend/UI Components

- edituser.html
 - Since the idea is to be able to add multiple classes to study for, there will be a document that allows a user to add another class to their profile. The user will also be able to edit or delete their profile from this view.
 - Requires:
 - studystyle.css, doj.js
 - Backend management and data parsing