Coding Buddy

Wyatt Lyda- Github: wyatt-lyda

Kaitlyn Brown- Github: TenorPanther

Ben Willcutt- Github: benwillcutt

Jack Toenjes- Github: Ponippp

Calvin Wagner- Github: CalvinWagner05

Christina Fernandez- Github: chfe8281

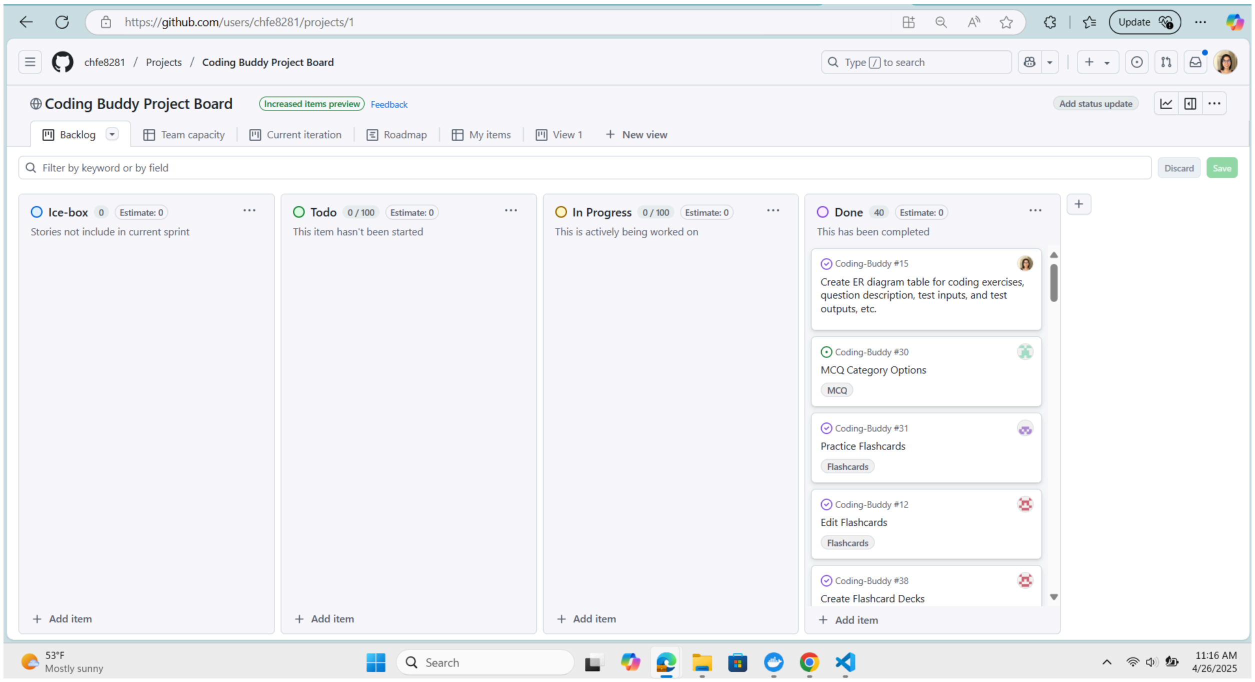
Project Description:

Coding Buddy is a new learning tool for computer science students at the University of Colorado Boulder. This app provides students with a customizable and approachable learning environment where they can challenge themselves. Rather than only completing coding questions, students have access to other learning materials to build up their knowledge before. Students can learn broad topics and conceptual material through built-in multiple choice quizzes and flashcards. They can then apply this knowledge to course-specific coding exercises. This would ideally relieve stress and prepare students more for the coding exercises, similar to those seen in exams or job interviews.

Students are led through a standard set of courses, including intro material, data structures, computer systems, and algorithms. Through this curriculum they will build a strong foundation in computer science. They are also shown a progress bar that depicts how far along they are in building this foundation.

Students can customize their learning by making their own sets of flashcards. If they choose to challenge themselves, they can solve more coding problems to earn more points and move up on the leaderboard. They could also time themselves while solving the coding problems and try to beat their own time.

Project Tracker - GitHub project board: https://github.com/users/chfe8281/projects/1/views/1



Ice-box 0 Estimate: 0
Stories not include in current sprint

Todo 0 / 100 Estimate: 0
This item hasn't been started

In Progress 0 / 100 Estimate: 0
This is actively being worked on

Done 40 Estimate: 0
This has been completed

Coding-Buddy #15
Create ER diagram table for coding exercises, question description, test inputs, and test outputs, etc.

Coding-Buddy #30
MCQ Category Options
MCQ

Coding-Buddy #31
Practice Flashcards
Flashcards

Coding-Buddy #12
Edit Flashcards
Flashcards

Coding-Buddy #38
Create Flashcard Decks

Video:

https://drive.google.com/file/d/18LoynxQ-lPo9L2sgCl_3DMajfmN7YLPp/view?usp=sharing

VCS:

https://github.com/chfe8281/Coding-Buddy.git

Contributions:

Christina:

At the beginning I helped set up the header and footer partials. I also set up the docker containers. Then, I contributed to the coding page. I spent some time researching APIs to compile the C++ code for the coding page until I found Piston. Then I made the post route and the passed partial to alert the user if they got the answer correct or not. Later, I made the coding timer and started the get route for the coding page. I worked with Wyatt to prevent repeated coding questions and lastly worked on styling the coding page.

Wyatt:

I contributed by implementing the login and register pages using Handlebars, JavaScript, HTML, and Bootstrap for styling. I also helped create partials for the header, footer, nav bar, and messages to ensure consistent layouts across the app. On the coding exercise page, I helped to implement topic buttons that generate random questions and added logic to detect when all questions in a course are completed, prompting a message and enabling practice mode. Additionally, I created algorithm-related coding questions for the coding exercise page. My work contributed to the authentication system, UI consistency, and enhanced interactivity of the coding exercises page.

Calvin:

My main assignment on the project was to create the home page for the website. This homepage contained information about many aspects of the project. It included details like our mission, how the website works, a To-Do section, and a leaderboard. On top of this, I also created the flip functionality for the flashcards. When you click on one of your decks, it takes you to a page that allows you to sift through the cards within said deck and flip them for a more engaging and interactive experience.

Kaitlyn:

I worked on creating the majority of the flashcards overview functionality. I created the SQL database, populated it with some default information like premade flashcards, and the html page with CSS and Handlebars to view the user's flashcards in a grid by their deck with a flip functionality in Javascript to see the front and back of cards. I also made modals to create, edit, and delete individual decks/flashcards with the constraint that only owners can edit/delete their decks/flashcards. I implemented the routes for these functionalities and another route for the user to add premade cards.

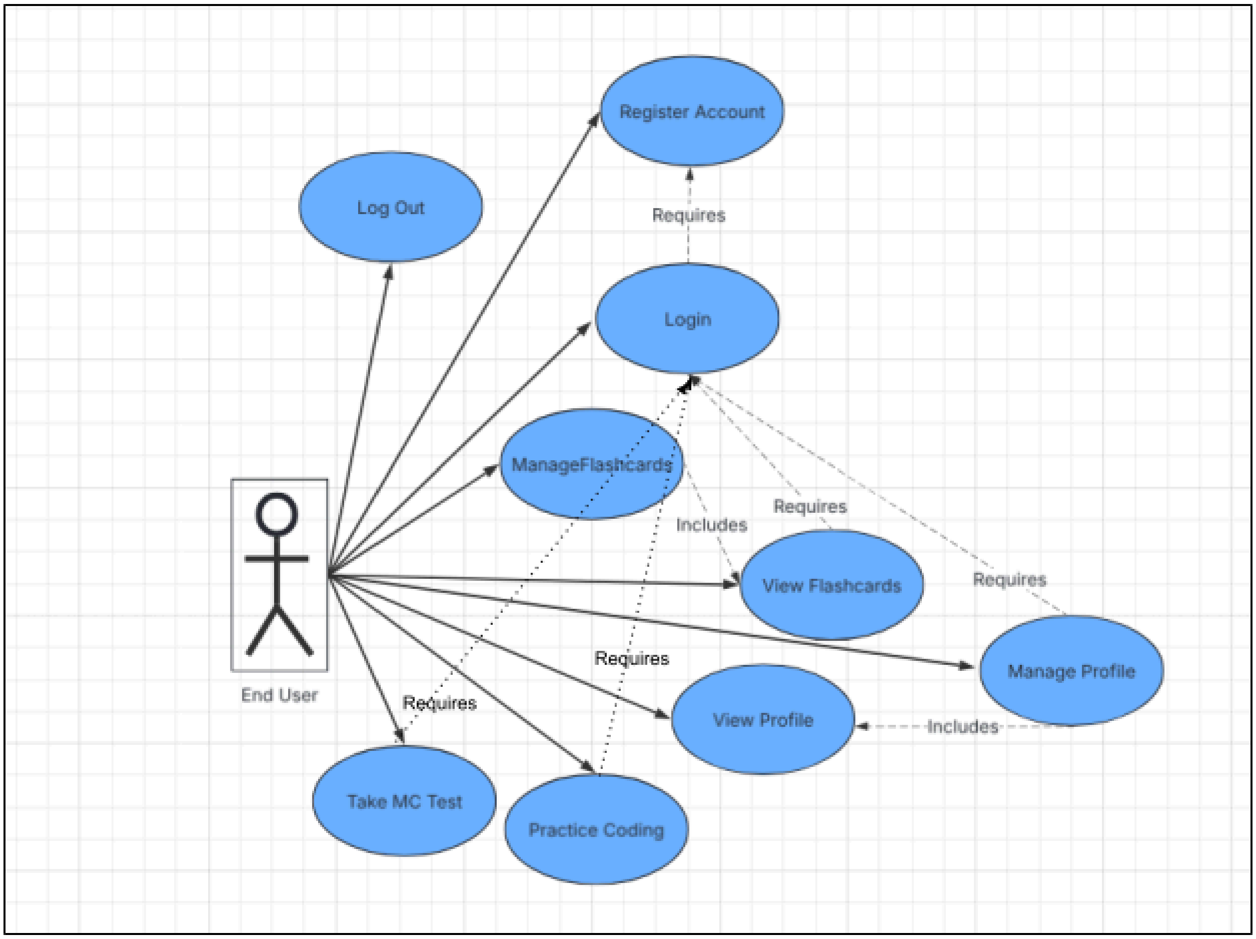
Jack:

I made the multiple choice quiz feature. I used html, hbs, and JavaScript. This feature includes the ability to check which answer is correct as the user goes through the quiz, the ability to select quizzes for four different courses, a 30s timer to respond, and the ability to receive points for correct answers that will display on your profile. I used Bootstrap and CSS to help style the page. Some stylistic choices I made include darkening the background color of the selected option, and changing the background color to green or red depending on the correctness of the checked option.

Ben:

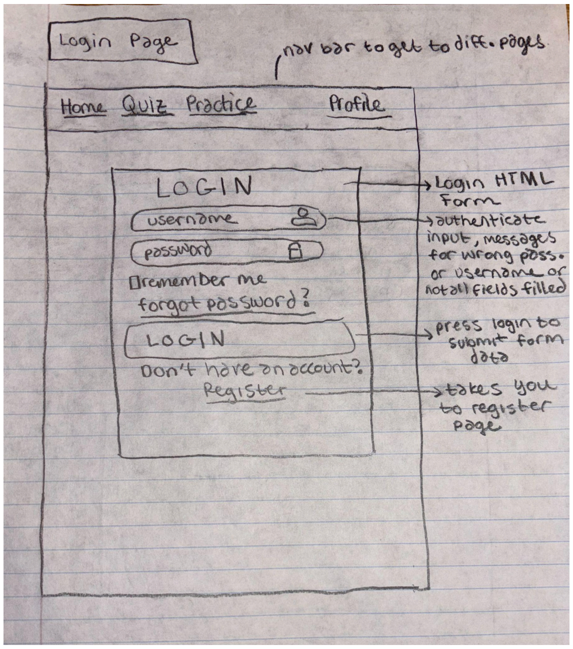
I created the profile, edit profile, and change password pages using html and bootstrap, allowing users to view their information, achievements, and update their details. I used JavaScript to implement a streak tracking feature to track how many consecutive days a user was actively interacting with the website and created a system to track how many points a user has earned from the coding exercises. On the coding exercises page, I built an autosave feature to prevent code loss if the user leaves the site or navigates to a different page. I also implemented testing using Mocha and Chai for the login, register, profile, multiple choice, and flashcards pages.

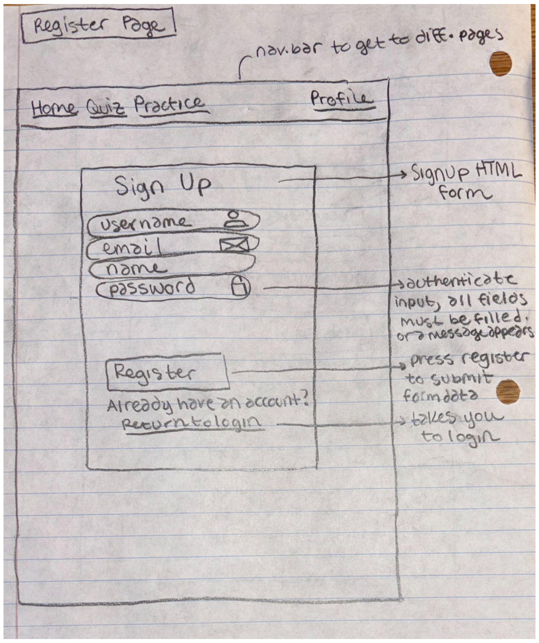
Use Case Diagram:



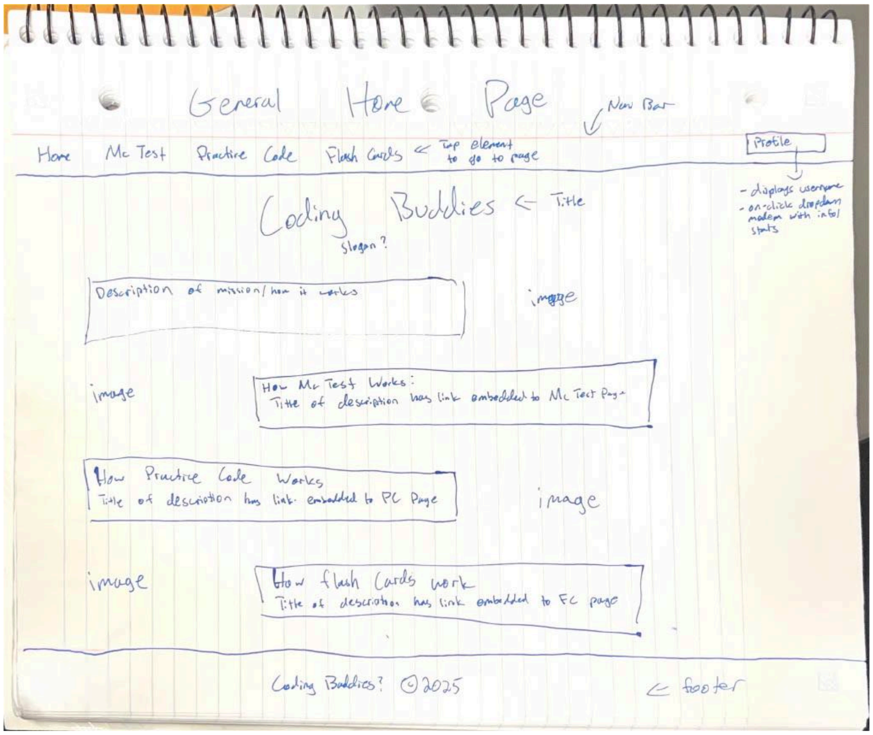
Wireframes:

Login/Register:

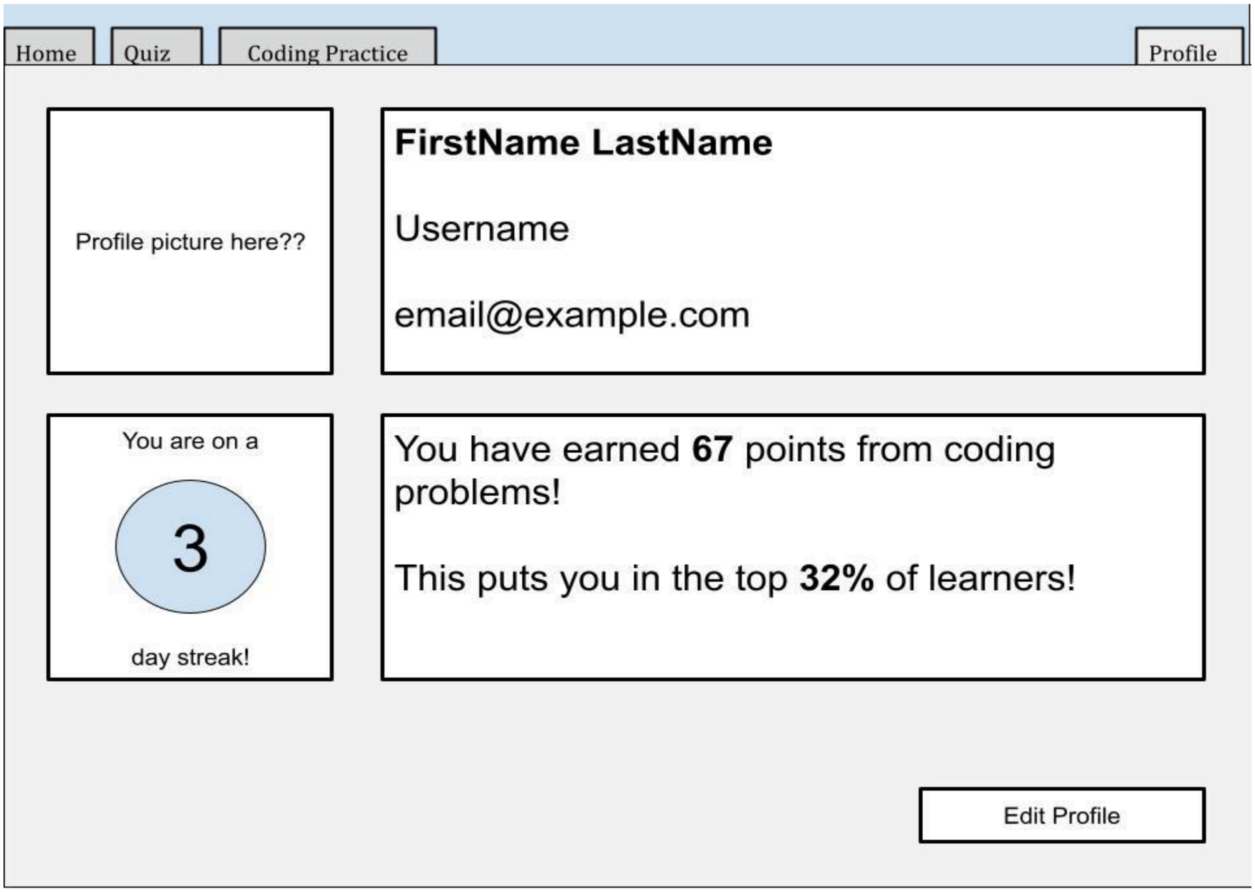

Login Page
nav bar to get to diff. pages
Home
Quiz
Practice
Profile
LOGIN
username
password
□remember me
forgot password?
LOGIN
Don't have an account?
Register
Login HTML Form
authenticate input, messages for wrong pass. or username or not all fields filled
press login to submit form data
takes you to register page



Home Page:


General Home Page
Nav Bar
Home
Mc Test
Practice Code
Flash Cards
<< Tap element to go to page
Profile
- displays username
- on-click dropdown modem with info/ stats
Coding Buddies
← Title
slogan ?
Description of mission/ how it works
image
image
How Mc Test Works:
Title of description has link embedded to Mc Test Page
How Practice Code Works
Title of description has link. embedded to PC Page
image
image
How flash Cards work
Title of description has link embedded to FC page
Coding Buddies? ©2025
← footer

Profile:



FirstName LastName

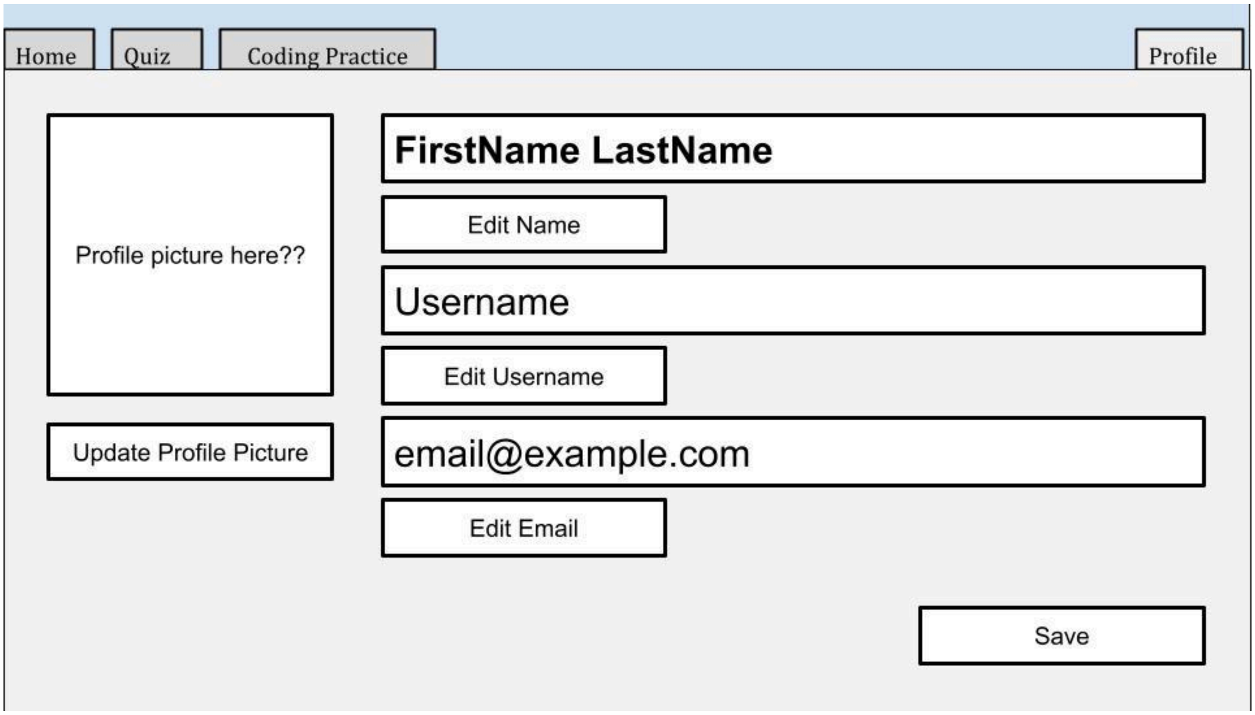
Username

email@example.com

You have earned **67** points from coding problems!

This puts you in the top **32%** of learners!

Edit Profile:

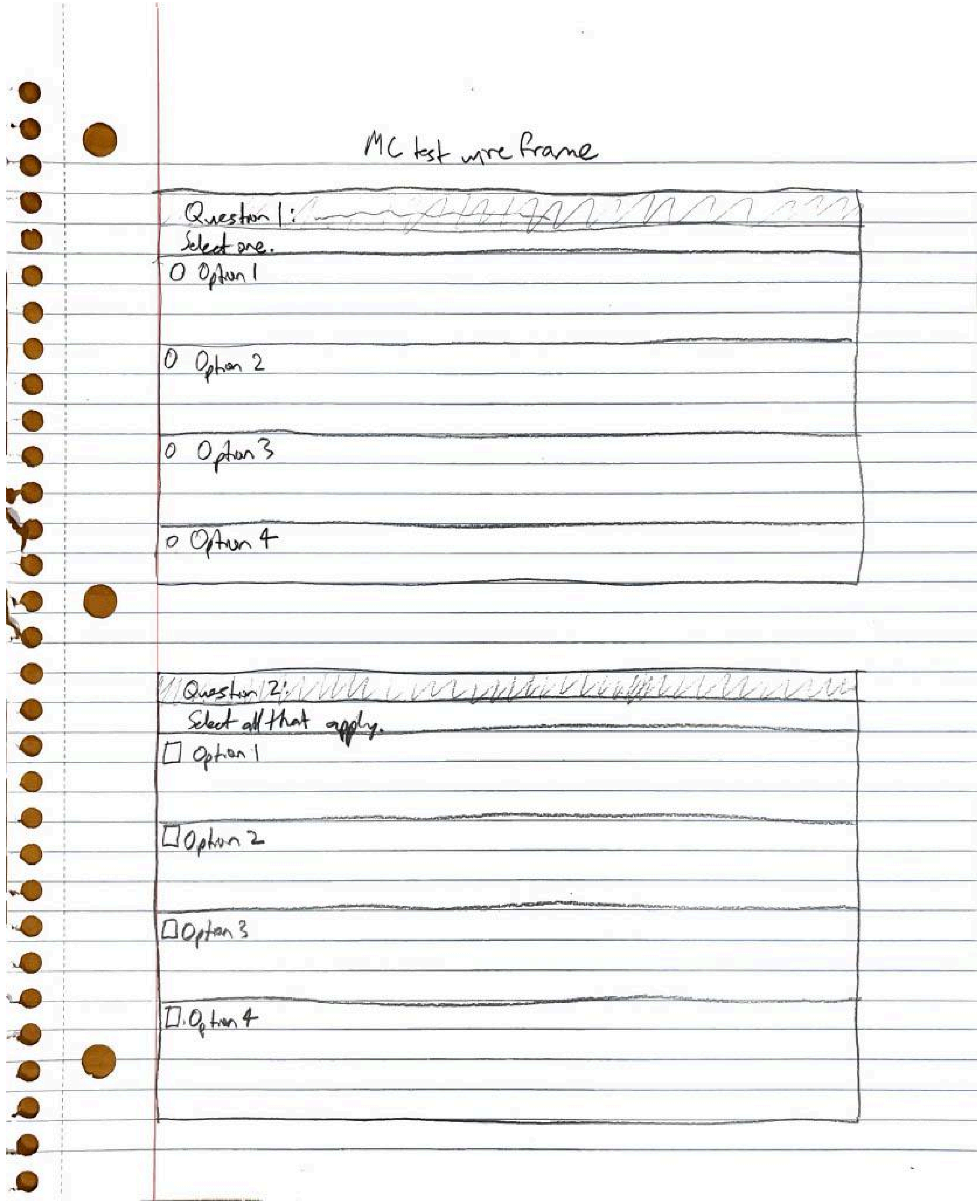


FirstName LastName

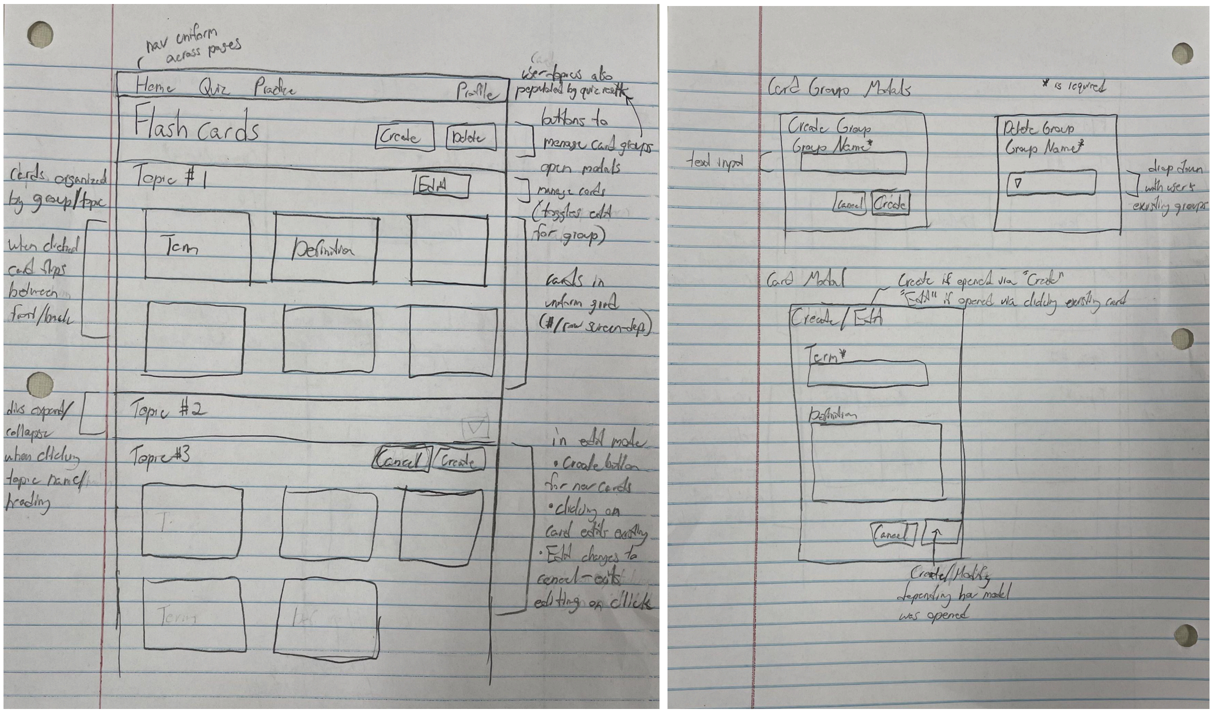
Username

email@example.com

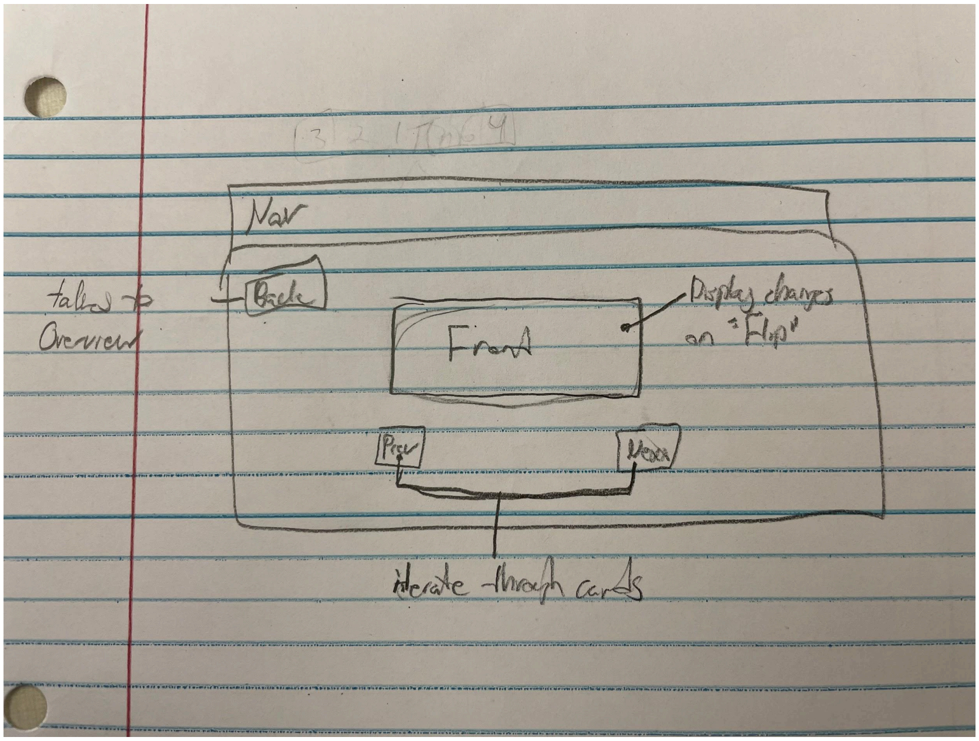
Multiple Choice Test:



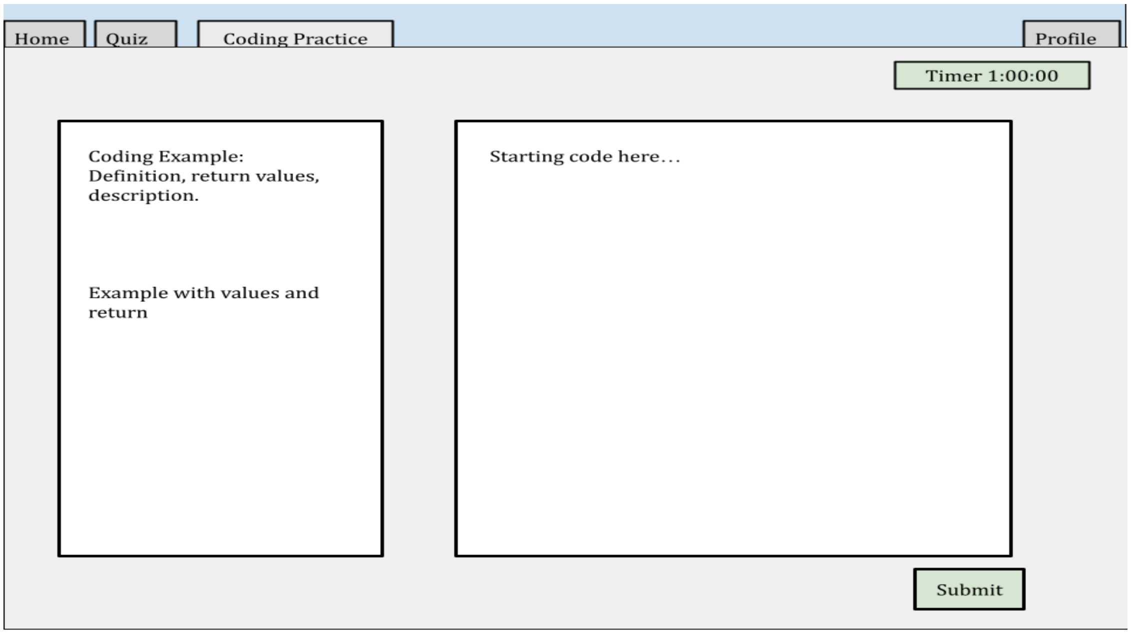
Flashcards Overview and Modals:


nav uniform across pages
Home
Quiz
Practice
Profile
Card user-topics also populated by quiz results
Flash cards
Create
Delete
buttons to manage card groups open modals
Topic #1
Edit
manage cards (toggles edit for group)
cards organized by group/topic
when clicked card flips between front/back
Term
Definition
cards in uniform grid (#/row screen-dep)
Topic #2
divs expand/collapse when clicking topic name/heading
Topic #3
Cancel
Create
in edit mode
• Create button for new cards
• clicking on card edits existing
• Edit changes to cancel - exits editing on click
Term
Card Group Modals
* is required
Create Group
Group Name*
text input
Cancel
Create
Delete Group
Group Name*
▽
drop down with user's existing groups
Card Modal
Create if opened via "Create"
"Edit" if opened via clicking existing card
Create / Edit
Term*
Definition
Cancel
Create/Modify depending how modal was opened

Flashcards Practice:



Coding Exercise:

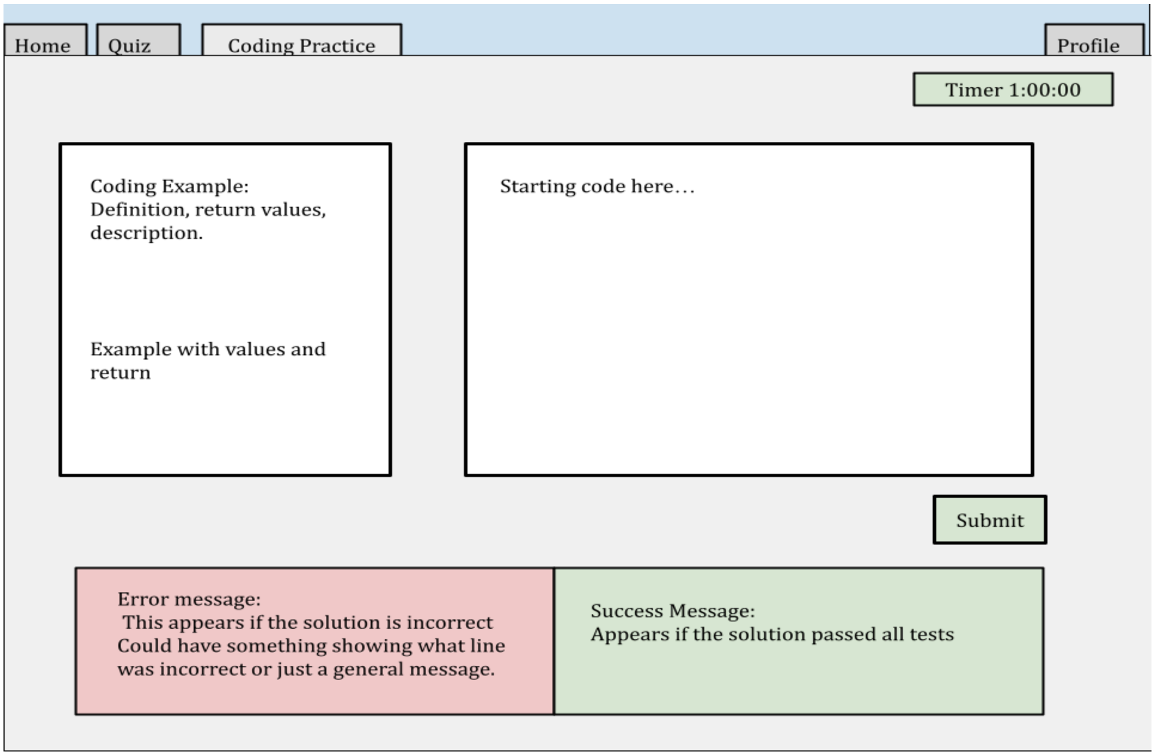


Coding Example:
Definition, return values, description.

Example with values and return

Starting code here…

Coding Exercise w/ message:



Coding Example:
Definition, return values, description.

Example with values and return

Starting code here…

Error message:
This appears if the solution is incorrect
Could have something showing what line was incorrect or just a general message.

Success Message:
Appears if the solution passed all tests

Test Results:

In Lab 11, we created a test plan for 3 of our pages: the register page, the login page, and the profile page. In our test plan, we initially said that we would be the testers for our website, but we decided that it would be beneficial to receive feedback from someone outside of our group, so we got a friend who wasn't in this class to interact with our website.

Register:

- For the register page, we wanted to make sure that the user could correctly input valid information to create an account on our website. Additionally, we wanted to check that if the user entered a username or email that was already taken, they wouldn't be able to successfully create an account.
- We observed our test user both successfully create an account with valid information and unsuccessfully create an account with invalid information, and everything worked as intended. The behavior was exactly what we expected for both the positive and negative cases, so we didn't have to change anything for this use case.

Login:

- For the login page, we wanted to make sure that the user could input correct information that they previously registered with and be successfully logged in to the website. We also wanted to check to make sure the user wouldn't be logged in if they entered a username that didn't match any existing user on our website.
- We observed our test user successfully log in with valid information and receive an error message when they tried to log in with an invalid username. The behavior was as expected for both cases, so we didn't have to adjust our application at all.

Profile:

- For the profile page, we wanted to make sure that the user could view all of their information and statistics on the page as well as potentially edit their name, username, email, or password if necessary.
- The test user was able to successfully view their information and stats on the profile page as well as update all of their information or change their password if they desired. We got feedback from our test user that the way the user's rank was displayed as a percentile was a bit confusing, so we ended up changing it to display the user's overall rank out of the total users on the website. Other than this small fix, everything else functioned as intended for the profile page.

In addition to these 3 use cases that we planned in Lab 11, we also carried out tests for the flashcards, multiple choice, and coding exercise pages.

Flashcards:

- For the flashcards page, we wanted to make sure that the user was able to create their own decks and cards as well as view the premade decks that we had created. We also wanted to ensure that the user couldn't access this page without being logged in.
- The test user was unable to access the page when logged out as intended. Once logged in, the user successfully clicked the button to get the premade flashcards and was able to create their own deck with flashcards. Everything on this page functioned as intended, so we didn't end up having to change anything based on the testing.

Multiple Choice:

- For the multiple choice page, we wanted to make sure that the user could view the quiz for each of the classes, get feedback for each question, and then see the score they got at the end of the quiz. Additionally, we didn't want the user to be able to access the page before logging in.
- The user couldn't access the page until they logged in, and then they were able to take each of the quizzes, receive feedback for each question, and then see their score at the end of each quiz. The user told us that it would be nice if the answer choice they selected was highlighted in some way, so we implemented this change following their feedback. Otherwise, the behavior was as intended for this page.

Coding Exercises:

- For the coding exercises page, we wanted to make sure that the user could enter code, submit their code, and then get feedback as to whether or not their solution was successful. We also wanted to make sure the user couldn't access this page until they logged in.
- The test user was unable to access the page when logged out as intended. Once logged in, the user could enter their code, click submit, and get a success message if their solution was accurate. If their solution was incorrect, they would get an error message that displayed the test cases in order to give them a hint as to where their solution went wrong. The one piece of feedback we received for this page was to potentially add an autosave feature so that the user's code would be saved if they accidentally refreshed or navigated away from the page. We were able to successfully implement this feature in the following days, so the user's code now autosaves. Otherwise, this page functioned as expected.

Deployment:

https://coding-buddy-e0hy.onrender.com