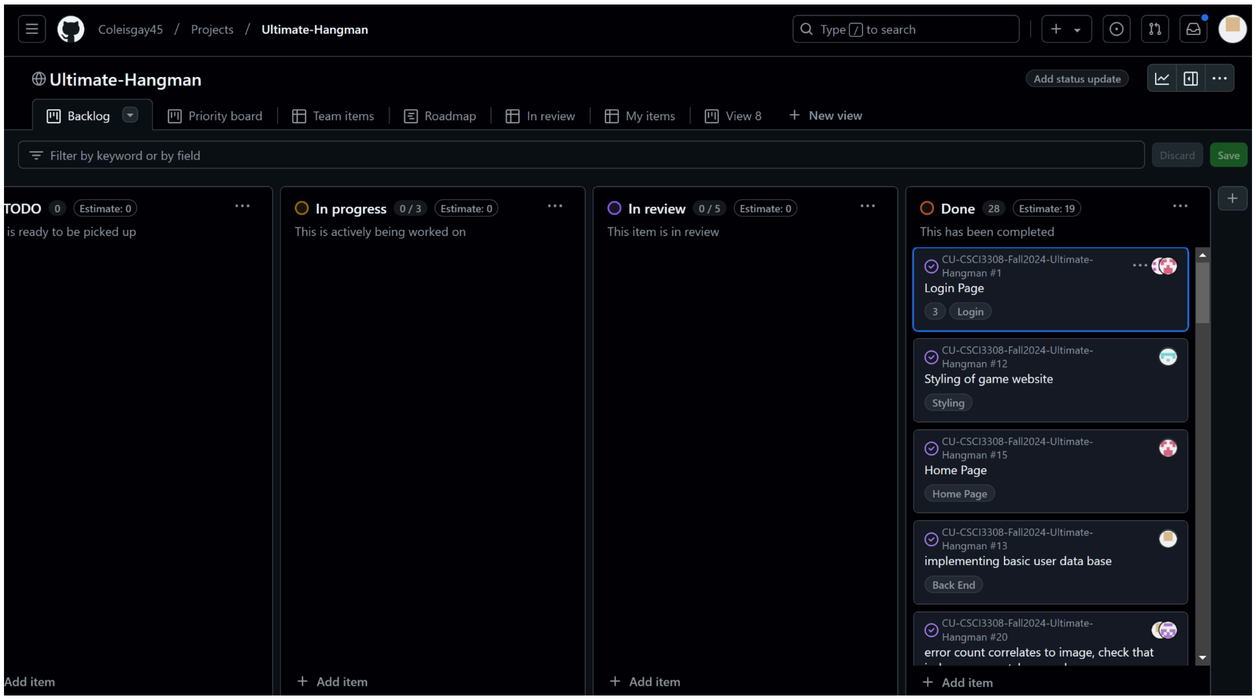
Project Name: Ultimate Hangman

People:
Dorrin Ashrafi, Ddelrayy
Tian Zhang, Hakuinu
Erasmia Poulassichidis, erpoulas
Cole Irish, Coleisgay45
Alex Recana, alexandriarecana

Project Description: Our project is a versatile and multi-faceted application that offers unique, wonderful, and engaging functionalities. Such features include a dictionary that utilizes an API to fetch and display information to users when they search a word, a unique take on the Hangman game with incredible display and engaging features, a database that stores user usage statistics and details, a word database featuring over 50k words, a leaderboard that lets you see the statistics of other users, along with other great features such as adjusting the theme of the display and difficulty settings. The versatility of our program will draw users from around the world with its ability to modify the display and difficulty to suit your personal needs. Ultimate Hangman is a revolutionary and innovative take on the Hangman Game and will captivate users with its amazing features and brilliant presentation.

Project Tracker - GitHub project board:

- Link to your Project Tracker (for instructor & TAs)
  - https://github.com/users/Coleisgay45/projects/2
- Screenshot showing your project in your project tracker



TODO 0 Estimate: 0
is ready to be picked up
Add item

In progress 0 / 3 Estimate: 0
This is actively being worked on
Add item

In review 0 / 5 Estimate: 0
This item is in review
Add item

Done 28 Estimate: 19
This has been completed

CU-CSCI3308-Fall2024-Ultimate-Hangman #1
Login Page
3 Login

CU-CSCI3308-Fall2024-Ultimate-Hangman #12
Styling of game website
Styling

CU-CSCI3308-Fall2024-Ultimate-Hangman #15
Home Page
Home Page

CU-CSCI3308-Fall2024-Ultimate-Hangman #13
implementing basic user data base
Back End

CU-CSCI3308-Fall2024-Ultimate-Hangman #20
error count correlates to image, check that

Add item

Video:

https://github.com/Coleisgay45/CU-CSCI3308-Fall2024-Ultimate-Hangman/blob/main/Final%20Submission/Screen%20Recording%202024-12-05%20at%201.44.54%20AM.mov

VCS: https://github.com/Coleisgay45/CU-CSCI3308-Fall2024-Ultimate-Hangman

Contributions:

Dorrin Ashrafi Soltanahmadi

I added a styled navbar with all buttons and linked the "Play Hangman" tab to playHangman.hbs. I parsed words from a text file into Easy, Medium, and Hard categories based on their length (Easy: 4-5 letters, Medium: 6-7 letters, Hard: >7 letters) and set Easy as the default difficulty. I wrote the backend endpoint for playHangman.hbs and created the initializeGame function in client.js to load the word and its definition when the game starts.I connected Erasmia's logic for word checking and blank display to my word setup and added a hint popup for definitions. I also debugged the login and registration endpoints to show messages, made wireframes for the hint and settings sections, gave ideas for the settings, and helped with the project presentation and report.

Erasmia Poulassichidis:

During planning, I made wireframes for the friend request feature, which we ultimately didn't implement, and the initial settings.hbs page. In development, I wrote all test cases. I wrote the game logic functions displayLetters and checkGuess, which Dorrin connected her features to. I contributed to making the keyboard connect to keyboard typing along with button click, and connected it so that it would run the checkGuess function and gray out/disable after being clicked. I helped Dorrin debug the hint pop up so it would connect across index.js, playHangman.hbs, and script.js. I debugged the registration endpoint after it was infinitely looping through utilizing my test cases. For the presentation, I made the architecture diagram and the future improvements slide.

Tian Zhang:

I created our word database by extracting words from a CSV file by using Python. I implemented the dictionary functionality of our project by making an API call by using the user's imputed word to an API. It returns and displays the word's definition as a popup message if the word exists or returns an error message if no such word exists. The other main functionality I implemented was the registration feature along with the endpoint. The user inputs a username and password. If the user does not exist, an endpoint call is made to insert the username and password (encrypted) into our user database. I also implemented the leaderboard feature. It displays every user and their easy, medium, and hard mode scores. Users are ranked by their hard mode scores.

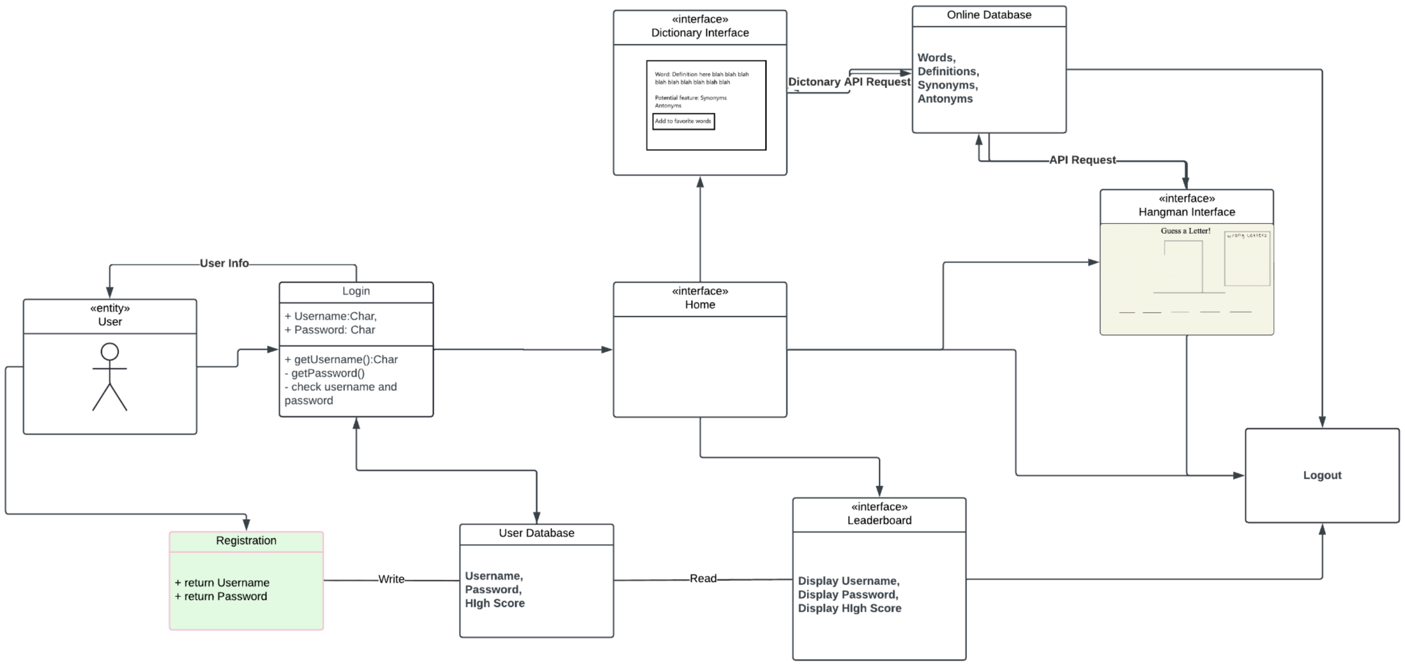
Alex Recana:

During our planning phase I created wireframes for the playHangman.hbs page. We ended up changing the layout of the page as we were developing it and I also created a gameOver.hbs page instead of displaying the user's result on the playHangman.hbs page. In development, I created and implemented the Logout page, Game Over page, and helped implement the Settings and Play Hangman page as well. I also contributed to the styling of our game with added color, themes, celebrations, and designing how the page looks. Working with Dorrin and Erasmia on the Play Hangman page, we found it rather difficult to implement, but we overcame the bugs and created a well-working game. I helped debug the issue of our theme (light and dark mode) display as we were having trouble displaying the theme to all pages of the website rather than just the settings page where the button was.

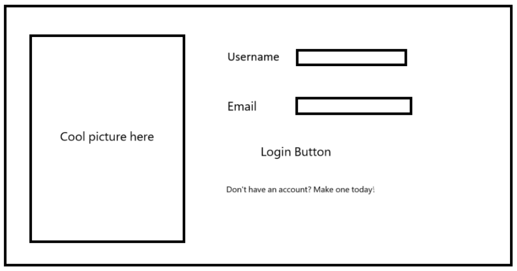
Cole Irish:

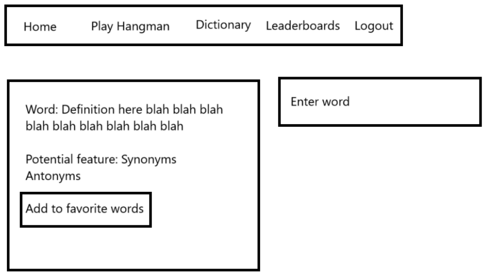
For our planning phase, I created a use case diagram for our application. For the development phase, I helped with the Login page and finished with the Login endpoint. I also debugged our connection with docker, as it was one of our earliest challenges. Then, I created the Home Page, as well as the endpoint for the page and all of the styling for the page. I created a live display of the users' scores on this page. I then started on the leaderboard page, with help finishing the page from Tian. I created the endpoint for this page, allowing live display of the scores of all the users. I next created a display on the Game Page, displaying only the score of the current difficulty the user is playing on. I finished with enabling scores to increase with each game won.

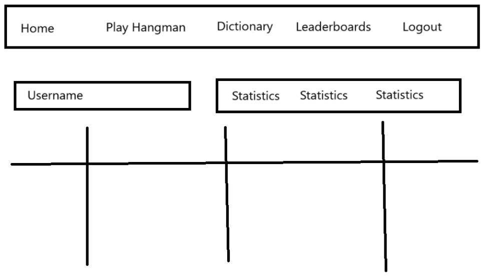
Use Case Diagram:

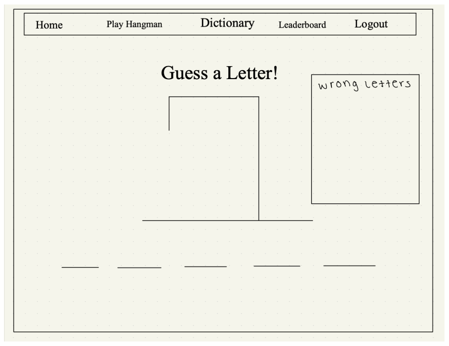

«interface»
Dictionary Interface
Word: Definition here blah blah blah blah blah blah blah blah blah
Potential feature: Synonyms Antonyms
Add to favorite words
Dictonary API Request
Online Database
Words,
Definitions,
Synonyms,
Antonyms
API Request
«interface»
Hangman Interface
Guess a Letter!
wrong letters
User Info
«entity»
User
Login
+ Username:Char,
+ Password: Char
+ getUsername():Char
- getPassword()
- check username and password
«interface»
Home
Logout
Registration
+ return Username
+ return Password
Write
User Database
Username,
Password,
HIgh Score
Read
«interface»
Leaderboard
Display Username,
Display Password,
Display HIgh Score

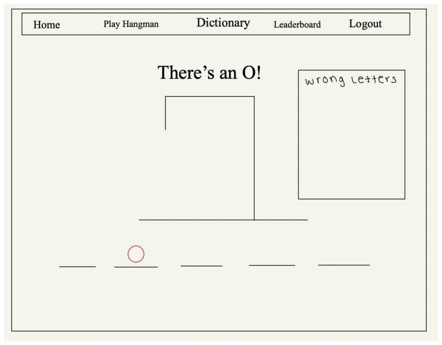
Wireframes:

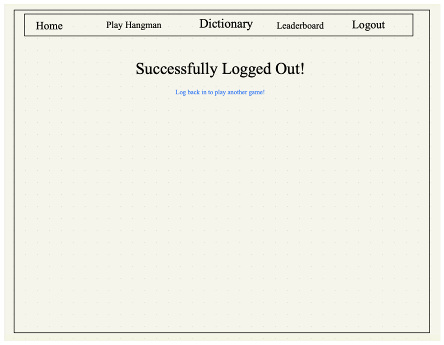

Cool picture here
Username
Email
Login Button
Don't have an account? Make one today!

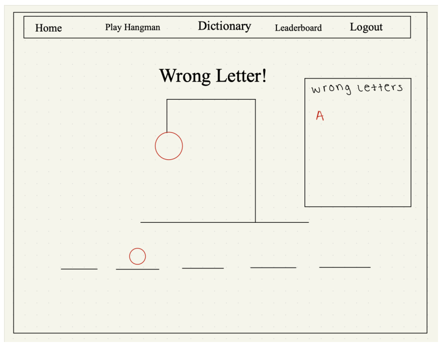

Home
Play Hangman
Dictionary
Leaderboards
Logout
Word: Definition here blah blah blah blah blah blah blah blah blah
Potential feature: Synonyms Antonyms
Add to favorite words
Enter word

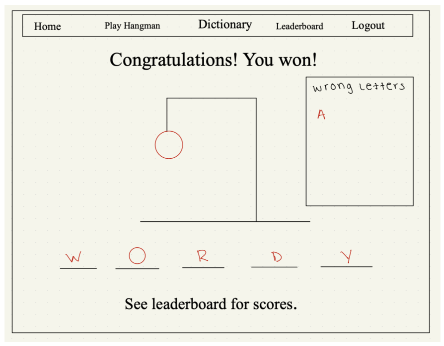

Home
Play Hangman
Dictionary
Leaderboards
Logout
Username
Statistics
Statistics
Statistics

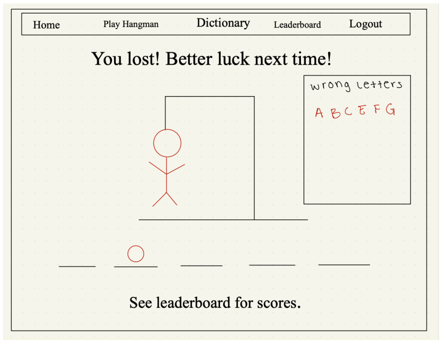

Home
Play Hangman
Dictionary
Leaderboard
Logout
Guess a Letter!
wrong letters


Home
Play Hangman
Dictionary
Leaderboard
Logout
There's an O!
wrong letters
O


Home
Play Hangman
Dictionary
Leaderboard
Logout
Successfully Logged Out!
Log back in to play another game!

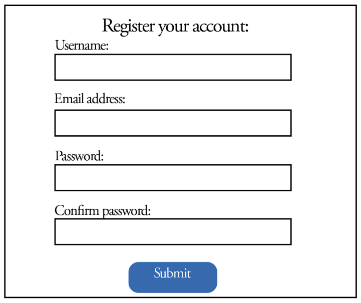

Home
Play Hangman
Dictionary
Leaderboard
Logout
Wrong Letter!
wrong letters
A
O


Home
Play Hangman
Dictionary
Leaderboard
Logout
Congratulations! You won!
wrong letters
A
W
O
R
D
Y
See leaderboard for scores.



You lost! Better luck next time!

See leaderboard for scores.



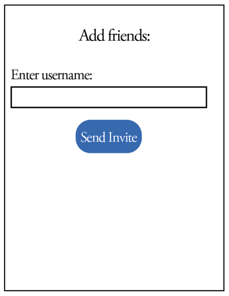
Register your account:

Username:

Email address:

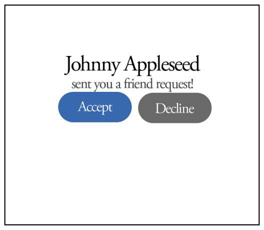
Password:

Confirm password:



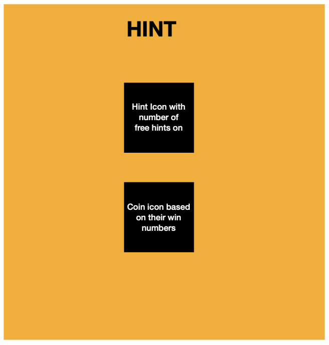
Add friends:

Enter username:



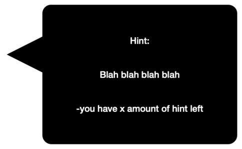
Johnny Appleseed

sent you a friend request!



HINT

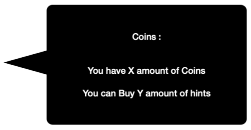
Hint Icon with number of free hints on
Coin icon based on their win numbers



Hint:

Blah blah blah blah

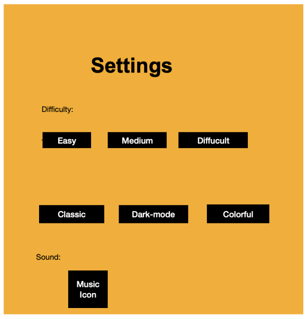
-you have x amount of hint left



Coins :

You have X amount of Coins

You can Buy Y amount of hints



Settings

Difficulty:

Sound:

Testing the dictionary feature:

A user should receive information concerning a word (i.e. word definition) when inputting and submitting a valid word

Test No.	Test Case	Steps	Test data	Expected Behavior	Results
WORD 001	Valid Word	1. Login 2. Click on the dictionary tab 3. Enter a word into the search bar 4. Click Search	User: Test Password: 1234 Input Word: Potato	User should be able to see the definition for the word potato	Definition of potato generated
WORD 002	Invalid Word	1. Login 2. Click on the dictionary tab 3. Enter a word into the search bar 4. Click Search	User: Test Password: 1234 Input Word: aeiofaiojcads	Should receive a blank field or a message that says No Results Found	Error message pops up at top of page

Testing the Hangman Game Feature:

The game should function as a typical game of hangman. If the user guesses a wrong letter / word, a part of the stickman hang appears and when all parts of the stickman appear, the game ends for the user and the user loses. If the user guesses the right word / letters, parts of the word will appear and if the user manages to fill in the entire word, the user will win and user statistics will be updated accordingly.

Test No.	Test Case	Steps	Test Data	Expected Behavior	Results
STICK001	Win Case	1. Login 2. Click On Play Hangman 3. Input Ward 4. Input Wurd 5. Input Word	Create a sample hangman game where the word is “word” and the computer guesses word in 3 tries Ward Wurd Word	User wins the game, i.e. a victory screen and user stats are updated accordingly to how many guesses it took	Victory screen generated, user score increases as seen in leaderboard.
STICK002	Lose Case	1. Login 2. Click on Play Hangman 3. Input Wurl 4. Input Wool 5. Input Wirl 6. Input Woof 7. Input Weel 8. Input Wall	Using the same sample hangman game as before where the sample word is “word”, the computer incorrectly inputs 6 incorrect inputs thereby losing the game	User loses the game i.e. a game over screen	Lose screen generated, user score remains the same.

Testing accept friend request function:

When a friend request pop up appears, if the user clicks accept friend request, the pop up goes away and the new friend is added to their friends list.

Test No.	Test Case	Steps	Test Data	Expected Behavior	Results
FRIEND001	Accept Friend Request	1. Log in as Testuser1 2. Click on User Search 3. Enter Testuser2 4. Click on Send Friend Request 5. Log out 6. Login as Testuser2 7. Click on Friend Invites 8. Click Accept Friend Request	Two Users Testuser1 and Testuser2, Testuser2 receives a friend request from Testuser1	After clicking on Accept Friend Request, Testuser1 and Testuser2 should be in both of their respective friend lists	Feature not implemented
FRIEND002	Reject Friend Request	1. Log in as Testuser1 2. Click on User Search 3. Enter Testuser2 4. Click on Send Friend	Two Users Testuser1 and Testuser2, Testuser2 receives a friend request from Testuser1	After clicking on Decline Friend Request, The friend request is rejected / discarded and the two users should not be in each others	Feature not implemented

		Request 5. Log out 6. Login as Testuser2 7. Click on Friend Invites 8. Click Reject Friend Request		friend list	

Login tests: As a user, upon entering valid login credentials, user should be logged in and have the homepage rendered for them.

Test No.	Test Case	Steps	Test Data	Expected Behavior	Results
LOG001	Successful Login	1. Input Logintest for the username field 2. Input pass123 for the password field 3. Click Login	User: Logintest Password: pass123	The User is successfully logged in and the home page is rendered for them	User enters session and can now navigate to other pages, their game data will be saved.
LOG002	Failed Login	1. Input Logintest for the username field 2. Input banana1 for the password	User: Logintest Wrong Password: banana1	An error message pops up saying wrong username or password	User is redirected to registration page, and error message pops up.

		field			

Deployment: Users should use this link, deployed using Render. If this link doesn't work, then users should run the application locally, pulling our application from github. To run the application locally, change the current path to CU-CSCI3308-Fall2024-Ultimate-Hangman\ProjectSourceCode. Then, with Docker open, in your Operating System's Shell Terminal type the following: docker compose up. In the internet browser of your choice, type localhost:3000 and then you can successfully run and use Ultimate Hangman!