PYTHON WEB APPLICATION GAME

HangMan

I decided to pre-plan the application by utilising the 5 planes of development as a planning framework.

Strategy

Brief:

- To design and build a web application version of the classic classroom game, Hangman.

Application Spec:

- Users should be able to play their own unique instance of the game.
- A user should be able to see their current score, and view historical top scores from other users.

User demographic:

- Children through to adults.

User needs:

- I want to be able to login using a username of my choosing.
- I want to be able to generate a random word which displays empty dashes depicting the words length.
- I want to be able to guess a letter, and have the result of that choice fed back to me.
- I want to know if I have guessed a word correctly, and how many points I get for that guess.
- I want to know if I haven't guessed a word correctly and what the answer was.
- I want to be able to see how well other users have done.
- I want to be able to logout when I am finished playing.

Scope

Feature set:

- Text input field to allow users to enter their username.
- A submit button to submit username.
- A generate button to retrieve a word to guess at random.
- Buttons or input for the user to submit their guesses.
- A div displaying the length of the word as dashes.

- Upon correct guess, the div of dashes should show the correctly guessed letter in place of the corresponding dash.
- Upon incorrect guess, a div displaying the stage of the “hangman” structure should increment.
- On correct guess of word, a div displaying the score obtained for the correct guess should display, and an overall running score of the current game should be incremented.
- On incorrect guess of word, a div displaying the word and a message of loss should be displayed.
- A link to a leaderboard page.
- A link allowing the user to logout of the game.
- A link taking the user back to the current game, from the leaderboard page.

Structure

Landing Page - First State

- Web application logo.
- Text input field.
- Submit/play button.

Game Page

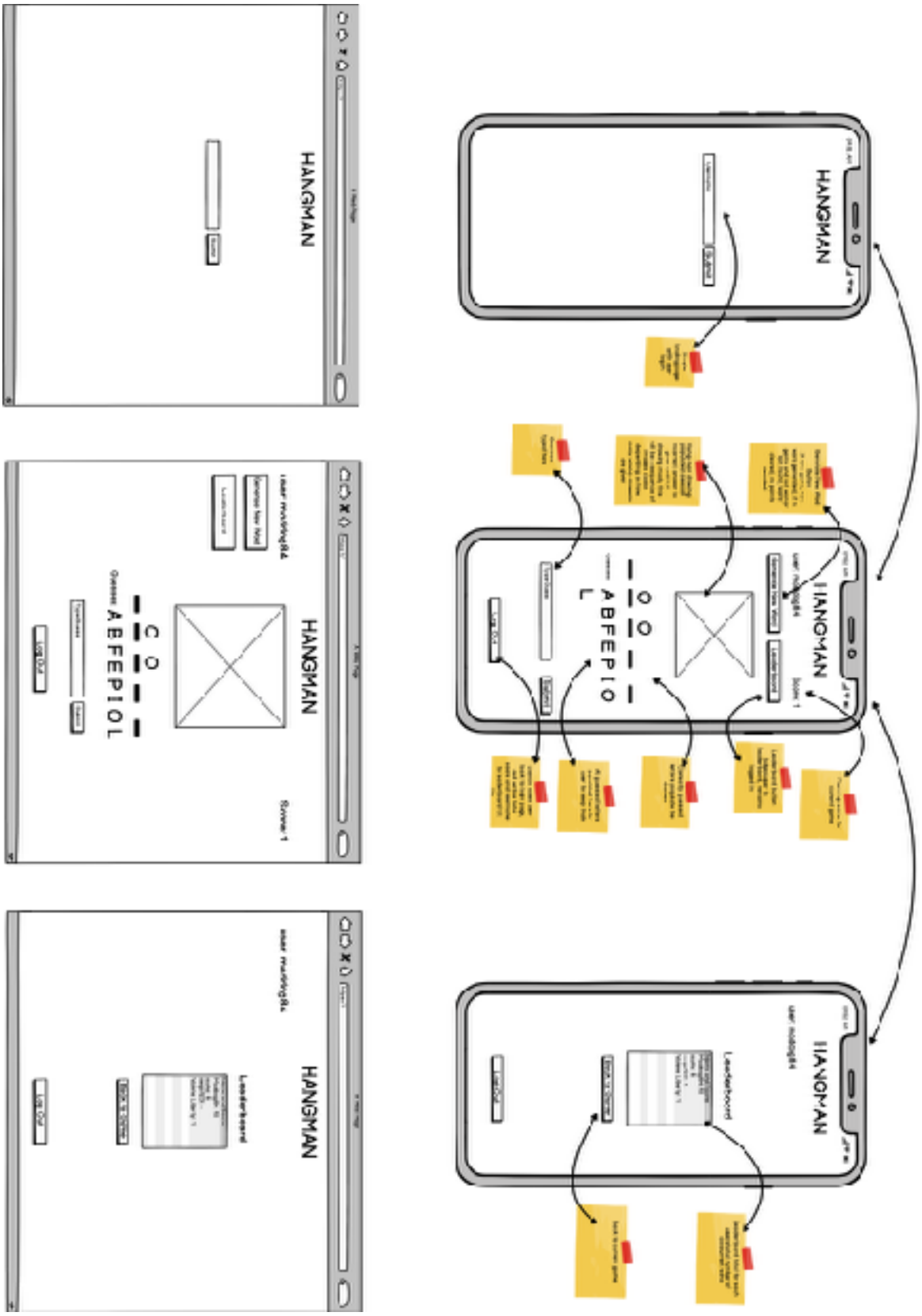
- Welcome message to user.
- Generate word button
- Nav bar taking user to Leaderboard and the ability to logout of the game.
- Buttons or input for guess submission
- Score counter
- Images displaying iteration of hangman structure.
- Result of guess.

Leaderboard Page

- Welcome message to user.
- Logout link.
- Back to game link.
- Table displaying top scores.

Skeleton

Using Balsamic I created wireframes, firstly developing the mobile layout, then considering from there how I'd like to present the same content on larger devices and desktops.



Surface

I wanted to reflect the origins of the hangman game when choosing styles for the application. For me clean and simple is always most effective. To achieve this, I went for a chalkboard background and two simple chalk based font styles. One for headings and one for text. Placing the game where it belongs, in the classroom.

I avoided styling any application features in a way that would take the user away from the illusion this game is taking place on a blackboard. Instead of styling buttons like modern CSS buttons, I opted to simply "bracket" a button's text to show that it's a button for user interaction. This way it looks like the button has been drawn on the blackboard, adding to the apps aesthetic rather than taking away from it.

The background image and fonts are free to use assets. The hangman structure images I created using illustrator.

I have used some simple CSS transitions to highlight which letters have been chosen, whether a button has been clicked or a link clicked. Again, as not to take away from the simplicity of the game's origins, I have kept JS and CSS trickery to a minimum.

Further Ideas

This application is a demonstration of my Python programming abilities in the context of a web based game. I have used other technologies such as HTML, SCSS, Jinja, Flask and Javascript to arrive at the finished result. Going forward, to strengthen the execution of the game there are some changes/improvements I would like to make.

1. Add password verification to the login page. Currently when a user logs out the game is over. If a user decided to return to their game, password verification would allow this to take place. Reinstating their score as they last left it.

2. Adding the ability to choose a category of words from which to generate a word would add another dimension to the game. Users could pick between topics such as historical figures, films or bands, artists or sports people etc.

3. Adding the ability to change the difficulty of the game. Giving the user the choice of short, medium or long words would again add another dimension to the game.