

# Project Two Title: A Brand New Project!

**Team Number: One**

**Team Name: The Collector (TBA)**

## Team Members:

|  |  |
| --- | --- |
| **Name** | **Email Contact** |
| Eleonore Caclard |  |
| Jason Grossman |  |
| Bill Geng |  |
| Sumant Singh |  |
| Neil Hall |  |

## Logo

**{Create a logo and place it here}**

## Project Board Website Link and Screenshots

**{Please place your link to your GitHub or Trello Project Board here}**

**{Place a baseline screenshot of your project board with your initial tasks }**

## 

## Project Description (High Level):

### Motivation

{ Describe why the team feels that the project idea can be an important commodity or service }

## User Story

{Describe below using the “As a [persona], I [want to], so that [outcome] “, to define the user story - see information for example at <https://www.atlassian.com/agile/project-management/user-stories> and detail the acceptance tests for the user story “Given…when…then” as described in <http://bit.ly/given-when-then> }

Example…

## Description

A Sneaker app made using node.js, express.js, mySQL, Sequelize ORM and handlebars.js that allows users to track their inventory of collected sneakers and socially interact with other sneakers collector

USER STORY

As a collector of quality Sneakers, I want to track my collection, add personal observations/comments to my catalogue, share my collection with others, and search for rare Sneakers to add to my collection.

ACCEPTANCE TEST FOR USER STORY

```md

GIVEN a Sneaker collection website

WHEN I visit the site for the first time

THEN I am presented with an animated landing page that prompt the user to login/register

THEN I am directed to the homepage/main page, which includes existing user posts if any have been posted, navigation links for the homepage and the dashboard

* The user is presented with the option to browse for collection
* 3rd part API to have options to view upcoming sneakers

WHEN I click on the homepage option

THEN I am taken to the homepage

~~WHEN I click on any other links in the navigation~~

~~THEN I am prompted to either sign up or sign in~~

~~WHEN I choose to sign up~~

~~THEN I am prompted to create a username and password~~

~~WHEN I click on the sign-up button~~

THEN my user credentials are saved, and I am logged into the site

WHEN I revisit the site later and choose to sign in

THEN I am prompted to enter my username and password

WHEN I am signed into the site

THEN I see navigation links for the homepage, the dashboard, and the option to log out

WHEN I click on the homepage option in the navigation

THEN I am taken to the homepage and presented with existing blog/comments posts that include the post title and the date created

WHEN I click on an existing blog/comment post

THEN I am presented with the post title, contents, post creator’s username, and date created for that post and have the option to leave a comment

WHEN I enter a comment and click on the submit button while signed in

THEN the comment is saved and the post is updated to display the comment, the comment creator’s username, and the date created

WHEN I click on the dashboard option in the navigation

THEN I am taken to the dashboard and presented with any blog posts I have already created and the option to add a new blog post

WHEN I click on the button to add a new blog post

THEN I am prompted to enter both a title and contents for my blog post

WHEN I click on the button to create a new blog post

THEN the title and contents of my post are saved and I am taken back to an updated dashboard with my new blog post

WHEN I click on one of my existing posts in the dashboard

THEN I am able to delete or update my post and taken back to an updated dashboard

WHEN I click on the logout option in the navigation

THEN I am signed out of the site

WHEN I am idle on the site for more than a set time

THEN I am able to view comments but I am prompted to log in again before I can add, update, or delete comments

```

GIVEN

Have both GET and POST routes for retrieving and adding new data.

Be deployed using Heroku (with data).

Use at least one new library, package, or technology that we haven’t discussed.

Have a polished UI.

Be responsive.

Be interactive (i.e., accept and respond to user input).

Have a folder structure that meets the MVC paradigm.

Include authentication (express-session and cookies).

Protect API keys and sensitive information with environment variables.

Have a clean repository that meets quality coding standards (file structure, naming conventions, follows best practices for class/id naming conventions, indentation, quality comments, etc.).

Have a quality README (with unique name, description, technologies used, screenshot, and link to deployed application).

NOTE: You may have more than one user story (or feature) to describe as part of your application.

## APIs to be Used:

{list the Names, descriptions and URLs of the APIs being used for your project below, along with the reasons on why you are using it }

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Resource | URL | HTTP Verb (GET, POST) | Action | Used For |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

## Libraries to be Used:

{List client-side CSS and JavaScript Libraries being used in this project. List the names, descriptions and URLs for these libraries, along with the reason to use it . Examples include JQuery, Materials UI}

CSS

Js

HTML

## Packages Required:

{Not Applicable for Project One}

 "dependencies": {

    "bcrypt": "^4.0.1",

    "connect-session-sequelize": "^6.1.1",

    "dotenv": "^8.2.0",

    "express": "^4.17.1",

    "express-handlebars": "^3.1.0",

    "express-session": "^1.17.0",

    "handlebars": "^4.7.3",

    "mysql2": "^2.1.0",

    "sequelize": "^5.21.5"

  },

  "devDependencies": {

    "jest": "^25.1.0"

  }

**+other npm package**

## Minimum Viable Product (MVP) Requirements

{Describe here your minimum features that are required to make your project functional and presentable}

* Add user 🡺extra
* Landing page
* Login (username, PW)

## Stretch Goals

{Describe here what features you would like to see if there is time permitting in your project period of development, or post-presentation, outline features that you would like to add for future development}

Adding other type footwear such as Louboutin, Manolo Blahnik, YSL, LVMH, …

Adding 3rd party API <https://tg4.solutions/the-sneaker-database-api-july-2020-release-notes/>

## Breakdown of Tasks (Ownership by Group Member):

{List who will be doing what}

Jason:API

Text

Description automatically generated with medium confidence

Eleonore:JS

Text

Description automatically generated

Neil:

Graphical user interface, application

Description automatically generated

Sumant: FE, landing page

Lodash

Nodemon

Request

Chai and mocha

Sam: Front-end design and development.

Jules: Game design and asset creation.

Kim: Back-end management, and integration with front-end.

Pat: Back-end management, schema creation, git repository management

Richie: Back-end design and development.

Kris: Front-end design and development.

Eddy: Back-end design, development, and integration with front-end.

Max: Game design and asset creation.

## Schedule for Completion of Tasks:

{Use a calendar to forecast estimated completion of stories and tasks, once they are defined. Have your teammates commit to it.}

|  |  |  |
| --- | --- | --- |
| **Date** | **Task** | **Notes** |
| **08/05** | Idea generation |  |
| **14/05** | Static pages for Main page, avatar, game, login/sign up page | Due on |
| **15/05** | Sample data - Main page, avatar, login/signup | Dynamic pages due on |
| **15/05** | Finalize and test GameSession class functionality | Due on |
| **16/05** | Game session | Dynamic pages due on & passed on for integration |
| **17/05** | Connect game logic to IndexedDB and graphql | Due on |
| **…** | … | **…** |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  | READY MVP |  |
|  |  |  |
|  |  |  |
|  | PROJECT POLISH |  |

## Schema

{ Describe your organization of your data used in the website, either in diagrams or text }

Example:

|  |  |  |  |
| --- | --- | --- | --- |
| Model | Properties / Fields | Type | Details |
|  |  |  |  |
| User | firstName | String |  |
|  | lastName | String |  |
|  | email | String |  |
|  | password | String (scrambled) |  |
|  |  |  |  |
| UserList | User (many) | Array (list) |  |
|  |  |  |  |
| Product | Name | String |  |
|  | Description | String |  |
|  | Price | Number |  |
|  | SKU | Number |  |
|  | ID | String |  |

Etc.

## Associations:

Example:

* A userList has many users
* ProductCategories has a list of individual Product Category information
* A Product Category has one or more products

Etc.

## Validations:

{Describe what forms will require validations, what fields within a form are being validated, and what type of validation is required}

Example:

New User Form

* Required Fields
  + First Name (text)
  + Last Name (text)
  + Email (valid email address)
  + Password (minimum, maximum, character type)

Search on Map Form

* Required Fields
  + City (text)
  + Province / State / District (text)
  + Country (text)

Etc.

## Migrations and Seed Information

{Describe here any initial data that you might need to initially apply (populate) before your website will work. For example , a list of “default users” }

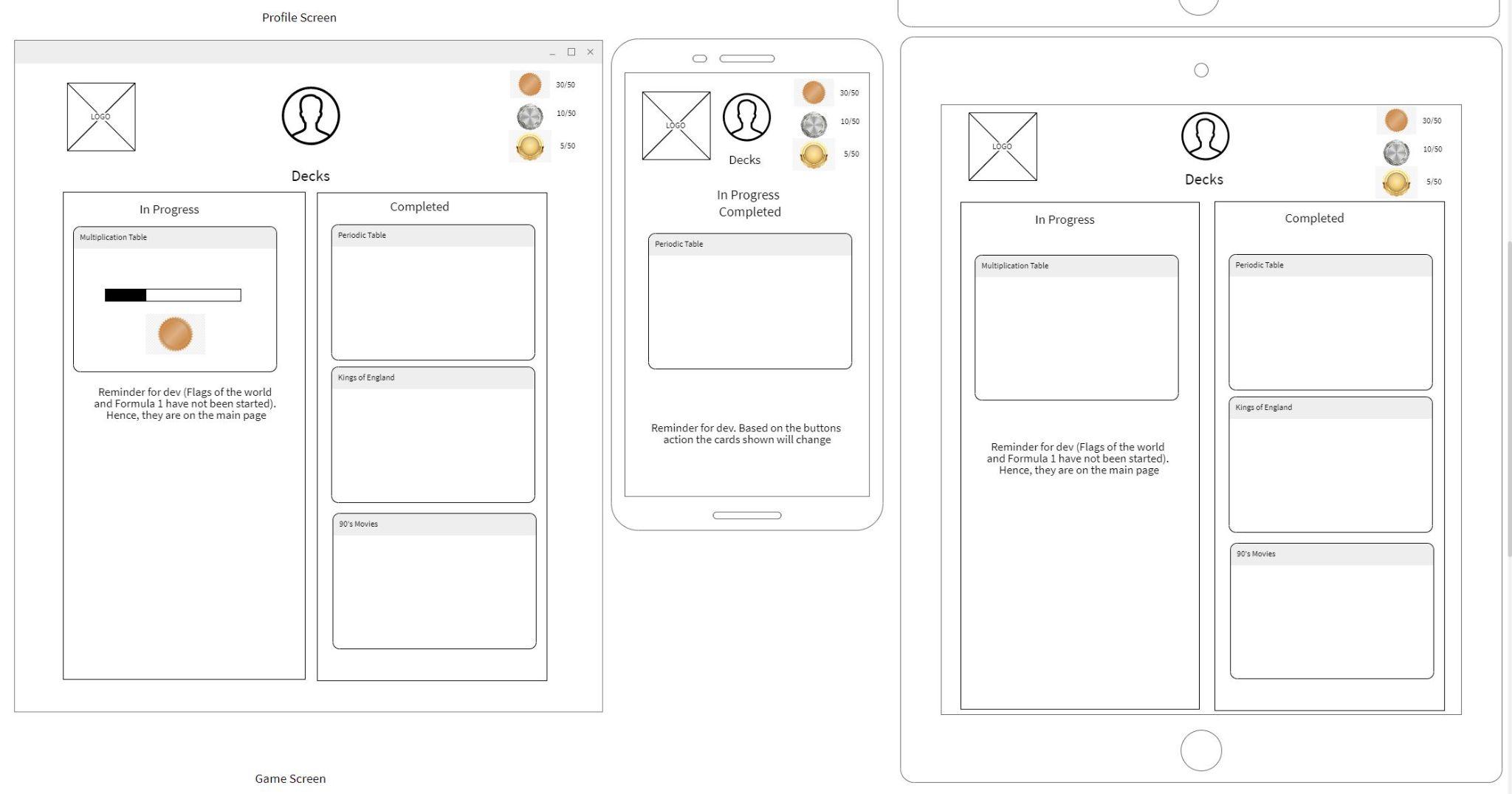
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| User | firstName | lastName | email | password | status |
|  | John | Doe | [john@doe.com](mailto:john@doe.com) | P@$$w0rd! | Active |
|  | Sally | Smith | [sally@smith.com](mailto:sally@smith.com) | T1geR\_1IlY | Inactive |
| … | … | … | … | … | … |

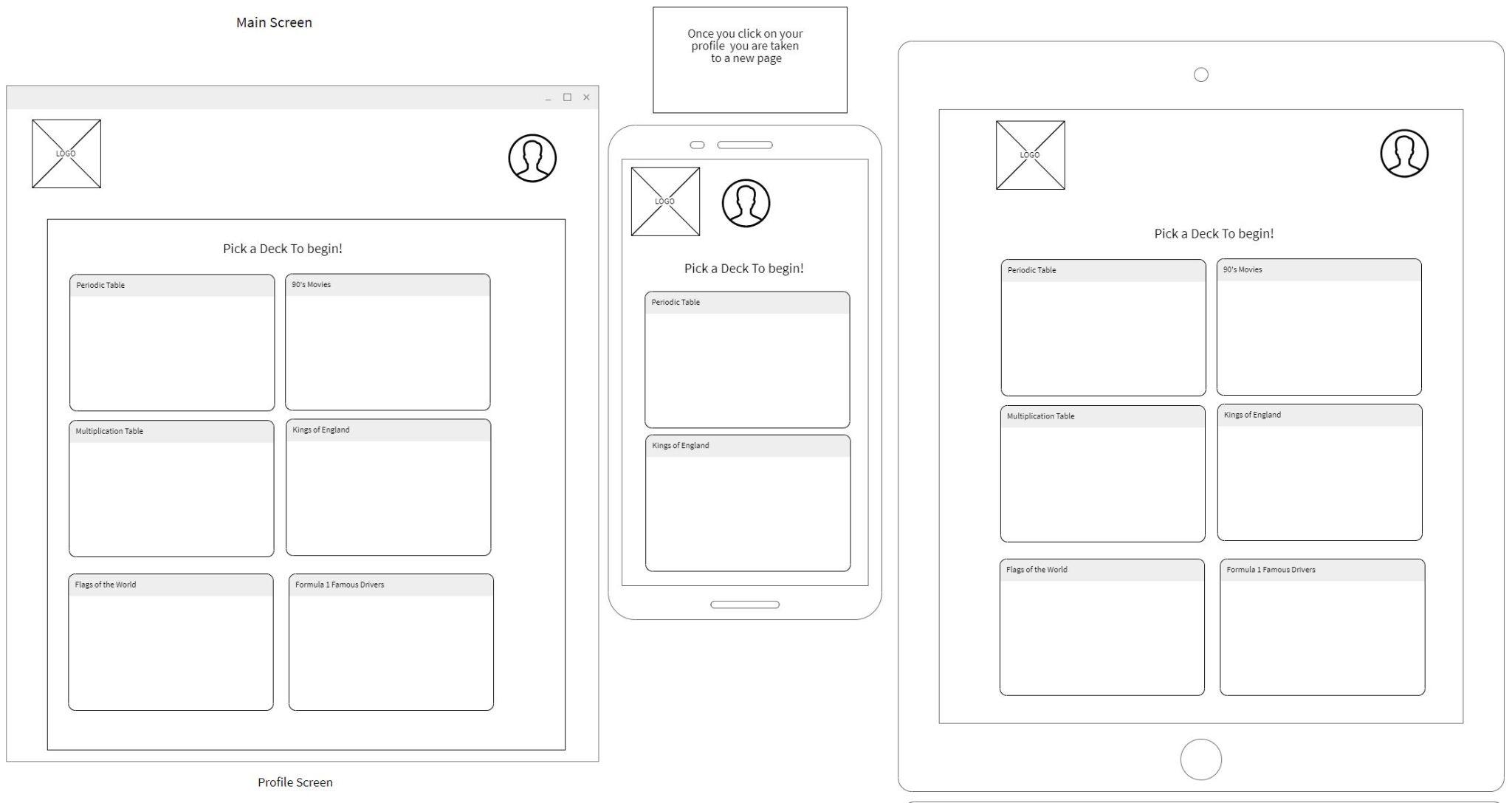
## Wireframes:

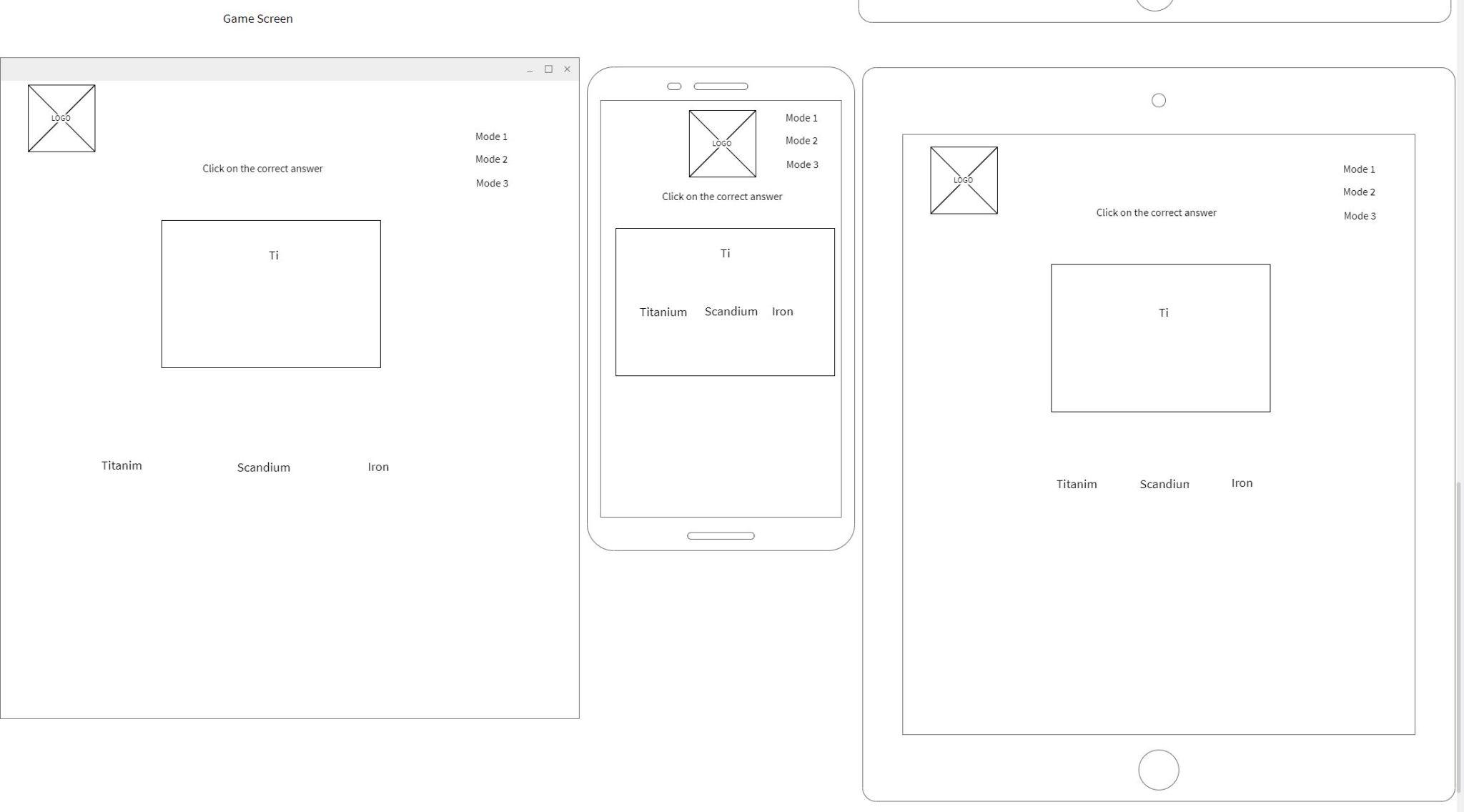
{ A set of DETAILED screen-by-screen design layouts with annotations describing all UI/UX components and all data relevant to the screen}

Acceptable formats:

* Scanned or photographed, neat and clean sketches, or
* UX/UI images created from tools such as balsamiq (<https://balsamiq.com>) , Lucidchart, PowerPoint, word, mockplus, Draftium etc.







## Website “Components” and/or “Sections”

{ Based on your wireframes, describe the common components on the website “navigation”, “forms”, “headers”, “footers” either with images / and or content and describe the intention / usage of the components. In addition, describe the primary “views” or “sections” that your project will have, for example “About Us”, “Products and Services”, etc.

}

## Git Workflow, Website Directory Structure:

{describe your workflow and methodology that all team members will abide by when using Git, AND describe in text or imagery the directories and filenames and their descriptions that will be used for the project}

Example:

### GIT Workflow

Production Branch Name: Main

Development Branch Name: develop

All developers will create branches off develop according to the naming convention:

{

example:

***feature/lastname/ticketnumber/task i.e. feature/Apostol/001/navigation-bar***

***docs/lastname/ticketnumber/task i.e. docs/Wayne/020/update-README***

***bug/lastname/ticketnumber/bug-desc i.e. bug/Kent/030/fix-navigation-bar-color***

}

### Code Review and Merge Requests

Identify Name(s) of people doing code review and approving Merge Requests: (lists names here)

* Team members should “pair up” to perform code reviews and merge requests
* At the halfway point, new pairs should be recreated
* If there is an odd number of participants in the team, then the extra member can float between different pairs to code review,

|  |  |  |
| --- | --- | --- |
| Week One | Team Member | Team Member |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
| Week Two |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

### Directory Naming and Structure; Filenames and descriptions

Only list the important files and folders.

|  |  |  |  |
| --- | --- | --- | --- |
| Directories | name | description | detail |
|  | images | Stores images |  |
|  | images/home | Images in home page |  |
|  | images/products | Product images |  |
|  | images/bio | Images for bio |  |
|  | css | Stores css | Main.css is principal stylesheet |
|  | css/nav | Navigation style |  |
| … | … | … | … |
|  |  |  |  |
| Files |  |  |  |
|  | Index.html | home |  |
|  | Contactus.html | Contact us page |  |
| … | … | … | … |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

## “Stuck Time” Agreement:

The team must identify and agree to a time frame that a team member must use as a limit before reaching out to (a) fellow team members and then (b) the instructional team (in-class) for support. That way, *no one should be stuck for ‘days’ or ‘many hours’ on a task.*

We the team members of (insert project team number or team name here) agree that the following period of time of investigation (with evidence) is allowed before identifying themselves as being “stuck” or “bottlenecked”, in which case, the team member can then seek the assistance of other team members, the instructional team, or AskBCS for assistance.

Period of time permitted before asking to seek assistance (hours): \_\_\_\_\_\_\_\_\_\_ hours

As agreed upon by the following team members (name and signature below)

|  |  |
| --- | --- |
| Name | Signature |
| Jane Doe |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

## Retrospectives – To Be Handed In At the Project Midpoint

Each team is to hold a *retrospective* meeting at the half-way point of the project period.

References:

* [What is a sprint retrospective - EasyRetro former FunRetro](https://easyretro.io/blog/what-is-a-sprint-retrospective/#:~:text=To%20facilitate%20this%2C%20the%20retrospective%20is%20based%20on,commit%20to%20doing%20differently%20in%20the%20next%20sprint%3F)
* [What is a retrospective? – All About Retrospectives (retromat.org)](https://retromat.org/blog/what-is-a-retrospective/)

Each team is to submit to the instructional team a short three paragraph Microsoft Word or PDF document at **the mid-point** of the project that describes the following:

* What worked well in the sprint?
* What didn’t work well and could be improved?
* What will we commit to doing differently in the next sprint?

## GitHub Repository Link, Production Website Link Submission Dates

To be filled in by the project team under advisement of the instructional team (and set out in Canvas)

* Presentation Date
* Project Submission Due Date
* Ideal Submission Date

## Additional Notes: