Hands-on Lab: Build Static Website Layouts

Estimated time needed: **90** minutes

Introduction

In this lab, you will build three **Medical Appointment Booking** website layouts, including the Navigation Bar, the Sign-Up form, and the Login form.

Objectives

After completing this lab, you will be able to:

- Create the layout and apply CSS styling for the following components:
    - Navigation Bar
    - Sign-Up form
    - Login form
- Test the layouts

Prerequisites

- You must have completed the prerequisite courses, especially the **Introduction to Web Development with HTML, CSS, JavaScript** course.
- You must have a basic knowledge of HTML and CSS.
- You must have completed the following labs:
    - Design Website Layouts
    - Create a GitHub Repository for your Project
- You must clone the forked repository in the Skill Network lab environment before you start this lab.

Project Scenario

Click **here** to review the project scenario.

Tips for finding and downloading images

Images form an important part of any website. The same applies to this Capstone project. You can either find your own images for the project or use these suggested images from *Pixabay* and *Unsplash*, sites that supplies royalty-free images.

- License information for Pixabay images
- License infromation for Unsplash images

After identifying the image, you can download the image using one of the following methods:

Using `wget` or `curl`

Use the `wget` or `curl` command to retrieve images from Pixabay and Unsplash. Make sure you `cd` to the correct folder so the files download to the appropriate directory.

Examples

```
wget https://cdn.pixabay.com/photo/2017/01/31/22/32/doctor-2027768_1280.png
```

```
curl -O https://cdn.pixabay.com/photo/2017/01/31/22/32/doctor-2027768_1280.png
```

Using the Cloud IDE Explorer

You might encounter situations when you need to upload images available on your local Desktop. In this case, you can use the Explorer available in the Skills Network Cloud IDE lab environment.

To upload the files to the relevant directory:

1. Click on the Explorer icon.
2. Navigate to the appropriate directory under the project.
3. On your local system, navigate to the folder where the images are present.
4. Select and drag one or more images from your local system and drop them in the appropriate project folder via Explorer.
5. Commit and push the content to GitHub.

    Refer to the **Capstone Project Reference: Git Commands** reading for details about Git command syntax and use relevant to the project.

Exercise 1: Create the Navigation Bar layout

1. From a new terminal, clone your forked repository and make sure that the Landing Page is available in the repository folder.

2. Set up the HTML structure for the Navigation Bar.

    1. Create a folder named **Navbar** in the **grihf-frontend_capstone_starter_code** folder.
    2. Create a new HTML file named **Navbar.html** in the **Navbar** folder and open it in a text editor in the lab environment.
    3. Start with the basic HTML structure by adding the `<!DOCTYPE html>` declaration and `<html>` tags.
    4. In the `<html>` tags, add the `<head>` and `<body>` tags.
    5. Add menu items for the Navigation Bar in a `div` tag within the `<body>` tag. Ensure these menu items are clearly visible and easily clickable.

3. The Navigation Bar must meet the following criteria:

    1. The Navigation Bar must include **Home**, **Appointments**, **Sign Up** and **Login**.
    2. Navigation Bar options must be laid out horizontally at the top of the page.

▶ Click here for the exemplar solution code for **Navbar.html**.

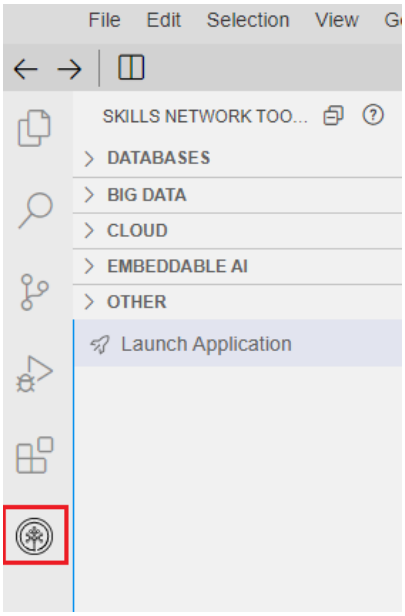
4. Apply CSS styling for the Navigation Bar. Make sure that you have linked **Navbar.css** in **Navbar.html** file.
    1. Create a CSS file named **Navbar.css** in the **Navbar** folder and open it in a text editor in the lab environment.
    2. Include styles for the following in the **Navbar.css** file:
        - Position of the Navigation Bar
        - Background color
        - Text spacing
        - Navigation links
    3. Make sure you have linked **Navbar.css** in **Navbar.html**.

▶ Click here for the exemplar solution code for **Navbar.css**.
▶ Click here for a **Navigation Bar** sample.

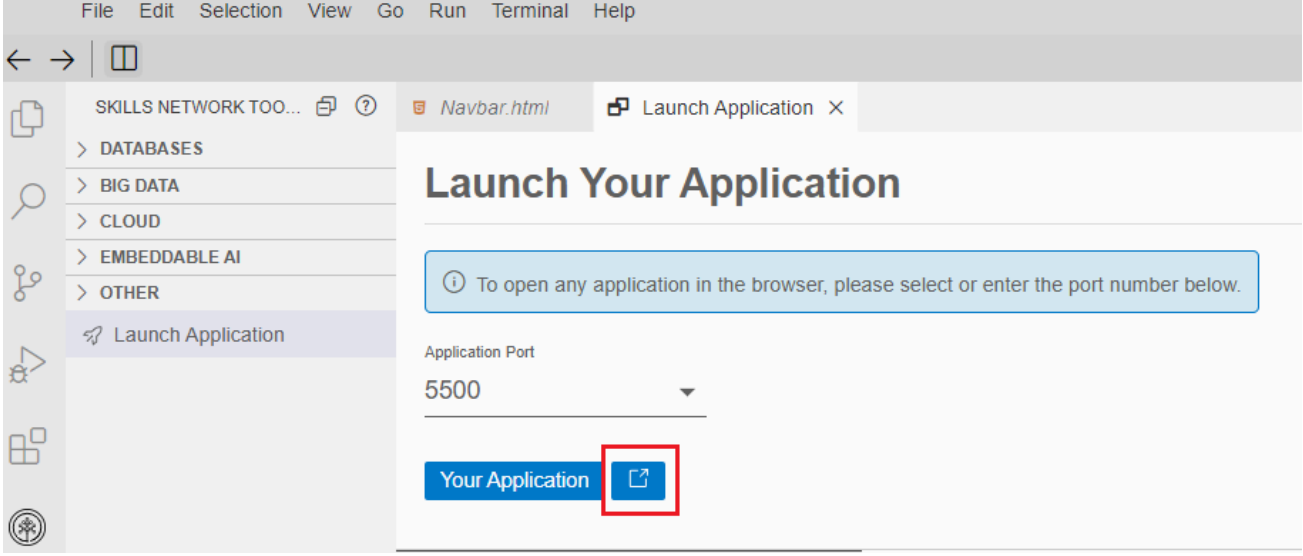
5. Next, you will need to check the functionality of the Navigation Bar. To do this:

    - Click **Go Live** at the bottom right under the terminal window in the lab environment.

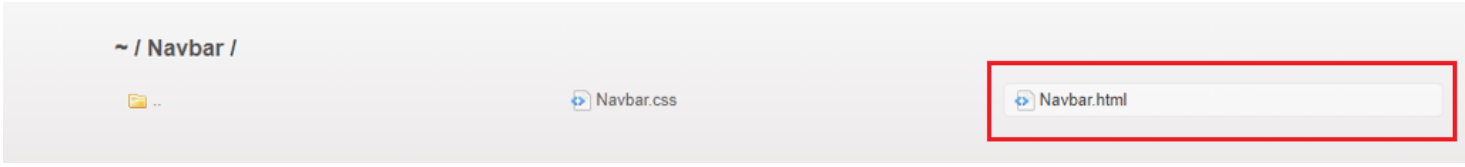
    - Then, from the Skills Network Toolbox, click **Launch Application**.



    - When prompted, enter the port number as 5500 and click the **Launch** icon.



- You will be redirected to **grihf-frontend_capstone_starter_code** in a new browser tab. Select the **Navbar** folder and then click **Navbar.html**.
- The Navigation Bar with all components must be displayed with any errors.



6. **Take a screenshot** of the Navigation Bar layout and save it as **navbar_layout.png**.

Exercise 2: Create the Sign-Up form layout

1. Set up the HTML structure for **Sign Up**.
    1. Create a folder named **Sign_Up** in the **grihf-frontend_capstone_starter_code** folder.
    2. Create a HTML file named **Sign_Up.html** in the **Sign_Up** folder and open it in a text editor in the lab environment.
    3. Start with the basic HTML structure by adding the **<!DOCTYPE html>** declaration and <html> tags.
    4. Create a `form` tag to add form elements including:
        - Role (for example, Doctor and Patient)
        - Name
        - Phone Number
        - Email
        - Password

       Note: Make sure to add both the input field and the label for each field.
    5. Add data validation to mark all fields as **Required** fields.
    6. Create a **Submit** button to submit the form data. You can also include a **Reset** button to allow a user to clear and re-enter all details.

▶ Click here for exemplar solution code for **Sign_Up.html**.

2. Apply CSS styling for the **Sign-Up** form. You must link the **Sign_Up.css** in **Sign_Up.html** file.
    1. Create a new CSS file named **Sign_Up.css** in the **Sign_Up** folder and open it in a text editor in the lab environment.
    2. Include styles for the position of the form, input fields, and labels in the **Sign_Up.css** file.

▶ Click here for exemplar solution code for **Sign_Up.css**.
▶ Click here for a sample output of the **Sign-Up** form.

5. Follow step 5 from exercise 1 to see the implementation of **Sign_Up.html** and **Sign_Up.css** file.
6. **Take a screenshot** of the Sign-up form layout and save it as **signup_form_layout.png**.

Exercise 3: Create the Login form layout

1. Set up the HTML structure:
    1. Create a folder named **Login** in the **grihf-frontend_capstone_starter_code** folder.
    2. Create a new HTML file named **Login.html** in the **Login** folder and open it in a text editor in the lab environment.
    3. Start with the basic HTML structure by adding the **<!DOCTYPE html>** declaration and <html> tags.
    4. Create a `form` tag to add form elements including:

- Email
- Password

5. Set email and password as **Required** fields.

▶ Click here for exemplar solution code for **Login.html**.

2. Apply CSS styling for the **Login** form. Make sure that you have linked **Login.css** in **Login.html** file.

    1. Create a CSS file named **Login.css** in the **Login** folder and open it in a text editor in the lab environment.

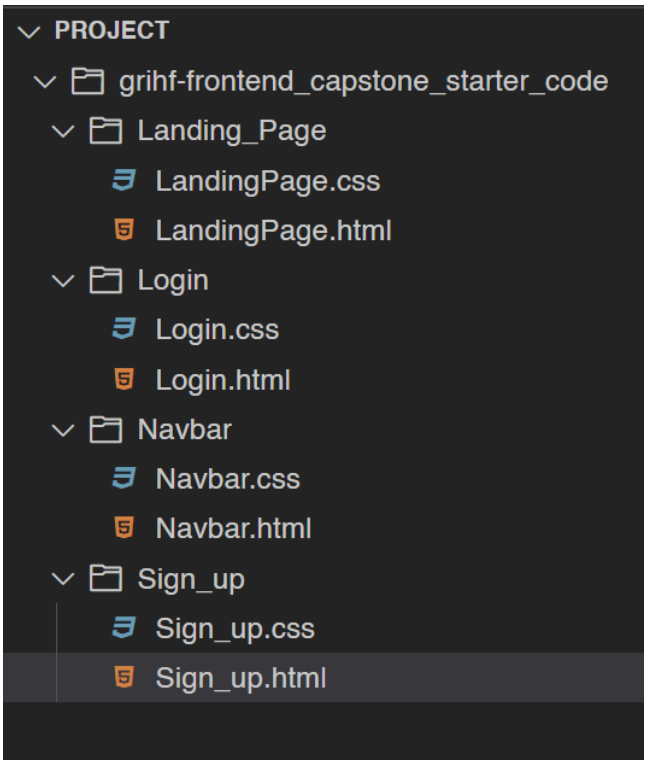
    2. Include styles for the position of the form, input fields, and labels in the **Login.css** file.

▶ Click here for exemplar solution code for **Login.css**.
▶ Click here for a **Login** form sample.

3. Follow step 5 from **Exercise 1** to launch the application and view Login.html. This will display the Login form's output.

4. **Take a screenshot** of the **Login** form layout and save it as **login_form_layout.png**.

    Note: All the folders should be maintained in a proper structure, with separate css and html files for code organization.



5. You must provide hyperlinks for the sign-up and login forms within the navigation menu items in the **Navbar.html** file, directing users to the respective Sign Up and Login forms. Additionally, establish reciprocal links between the Sign Up and Login forms. Verify the operational accuracy of these links to ensure proper functionality.

6. Perform `git config --global` for name and email, `git add`, `git commit`, and `git push` commands to update changes into your React project's GitHub repository for proper code management.

    Refer to the **Capstone Project Reference: Git Commands** reading for details about Git command syntax and use relevant to the project.

Screenshot checklist

You should have taken the following screenshots as part of this lab:

- *navbar_layout.png*
- *signup_form_layout.png*
- *login_form_layout.png*

Note about data management and persistence

To ensure the proper management and persistence of your data in a GitHub repository, it is crucial to follow a few essential steps:

- **Regular updates:** Whenever you make changes or add new components to your project, it is essential to add, commit, and push the updates to your GitHub repository. This ensures that your latest work is safely stored and accessible to collaborators.
- **Session persistence:** During an active session, your data remains accessible. However, it's important to note that if your session expires or you log out, you will need to clone the repository again to resume work.
- **Ignoring node modules:** When pushing data to GitHub, it's best practice to exclude the node modules folder from both your server and client directories. This folder contains external dependencies and can be quite large, making the repository heavy and slowing down the process. By adding it to the .gitignore file, you prevent it from being pushed to the repository, keeping your commits cleaner and more focused.

By adhering to these guidelines, you can maintain a well-organized and efficient GitHub repository, ensuring that your work is securely stored and easily accessible to you and your collaborators.

Author(s)

Richa Arora