# UNIVERSITY OF DELHI

## KALINDI COLLEGE

*Submitted To: Mr. KULANSHU SHARMA*

*Submitted By: NIDHI MISHRA*

*Course: B.VOC (WEB DESIGNIG)*

*Roll No.: 22105073*

*Subject: Full Stack Web Development*

# INDEX

| S.NO | TOPIC | PAGE NO. |
|---|---|---|
| 1. | HTML, CSS & JavaScript Introduction | 03 |
| 2. | About API works and Integration | 04 |
| 3. | About the Project | 05-06 |
| 4. | Implementation of Project | 07 |
| 5. | Screenshot of working Project | 08 |

# Chapter 1: INTRODUCTION

## 1.HTML (Hypertext Markup Language):

HTML provides the structure and content of web pages. In the "Know Your World" project, HTML is used to define the layout of the web application, including elements such as headings, paragraphs, lists, and forms. HTML tags are utilized to structure the user interface and organize content in a semantic and accessible manner.

## 2.CSS (Cascading Style Sheet):

CSS is used to style and layout the HTML elements of the web application. In the "Know Your World" project, CSS rules are applied to define the visual presentation, including colors, fonts, margins, and positioning. CSS enables the customization of the user interface to create an appealing and cohesive design.

## 3.JavaScript:

JavaScript is used to add interactivity and dynamic behaviour to the web application. In the "Know Your World" project, JavaScript is employed to handle user interactions, make API calls, retrieve and manipulate data, and update the DOM (Document Object Model) in response to user actions.

By incorporating HTML, CSS, and JavaScript into the "Know Your World" project, we create a dynamic and engaging web application that provides users with comprehensive country information in an intuitive and visually appealing manner.

# Chapter 2: About API works and Integration

1. **Positions API Integration:**
    - **Endpoint:** The Positions API provides latitude and longitude positions of countries.
    - **Integration:** When the user selects a country and clicks on the "Location" button, the application fetches data from the Positions API using a GET request.
    - **Data Retrieval:** The fetched data includes latitude and longitude coordinates, which are then dynamically displayed on the webpage.

2. **Currency API Integration:**
    - **Endpoint:** The Currency API retrieves currency information of countries.
    - **Integration:** Upon user interaction with the "Currency" button, the application sends a GET request to the Currency API endpoint.
    - **Data Processing:** The received data contains currency details such as currency name. The application extracts this information and presents it on the webpage.

3. **Capital API Integration:**
    - **Endpoint:** The Capital API provides capital city information of countries.
    - **Integration:** When the user clicks on the "Capital" button, the application makes a GET request to the Capital API endpoint.
    - **Data Extraction:** The retrieved data contains the name of the capital city for the selected country. This data is then displayed on the webpage.

4. **Population API Integration:**
    - **Endpoint:** The Population API retrieves population data for cities within countries.
    - **Integration:** Upon user interaction with the "Population" button, the application sends a GET request to the Population API.
    - **Data Parsing:** The received data includes population details for various cities within the selected country. The application extracts this information and formats it for display on the webpage.

## INTEGRATION DETAILS:

- **Fetch API:** The Fetch API is used to make asynchronous HTTP requests to the API endpoints.
- **Response Handling:** Upon receiving the response from each API, the application parses the data into JSON format.
- **Data Extraction:** The relevant information is extracted from the API response based on the user-selected country.
- **Dynamic Content:** The extracted data is dynamically inserted into specific HTML elements on the webpage, allowing users to view the country-related information.

# Chapter 3: About The Project

1. **Project Features:**
   - **User Input Validation:** Ensures that the user enters a valid country name before proceeding.
   - **Data Retrieval:** Fetches various country-related data such as positions, currency, capital, and population from external APIs.
   - **Dynamic Content:** Displays the fetched data dynamically on the webpage based on user interactions.
   - **Interactive Interface:** Provides buttons for users to select the type of information they want to retrieve.
   - **Error Handling:** Handles errors gracefully, ensuring a smooth user experience even in case of API failures.

2. **Technologies Used:**
   - **Frontend:** HTML, CSS, JavaScript (DOM Manipulation)
   - **External Libraries:** Fetch API (for making HTTP requests), localStorage (for storing user-selected country).

3. **Project Structure:**
   - **HTML Files:** Define the structure of the web pages, including input fields and placeholders for displaying data.
   - **CSS Files:** Styles the elements on the page for an attractive and consistent user interface.
   - **JavaScript Files:** Implements the application's logic, including user input validation, API integration, and DOM manipulation.
   - **API Endpoints:** External APIs are accessed to retrieve specific data about countries.

4. **API Endpoints:**
   I. **Position API:**
      - **Method:** POST
      - **Endpoint:** "https://countriesnow.space/api/v0.1/countries/positions"
      - **Purpose:** Retrieves the latitude and longitude positions of countries.
      - **Usage:** Used to fetch the position data for a specific country.

   II. **Currency API:**
      - **Method:** GET
      - **Endpoint:** "https://countriesnow.space/api/v0.1/countries/info?returns=currency"

- **Purpose:** Retrieves the currency information of countries.
- **Usage:** Used to fetch the currency data for a specific country.

III. **Capital API:**

- **Method:** GET
- **Endpoint:** "https://countriesnow.space/api/v0.1/countries/info?returns=capital"
- **Purpose:** Retrieves the capital city information of countries.
- **Usage:** Used to fetch the capital city data for a specific country.

IV. **Population API:**

- **Method:** POST
- **Endpoint:** "https://countriesnow.space/api/v0.1/countries/population/cities"
- **Purpose:** Retrieves population data for cities in countries.
- **Usage:** Used to fetch population data for cities within a specific country.

# Chapter 4: Implementation of the Project

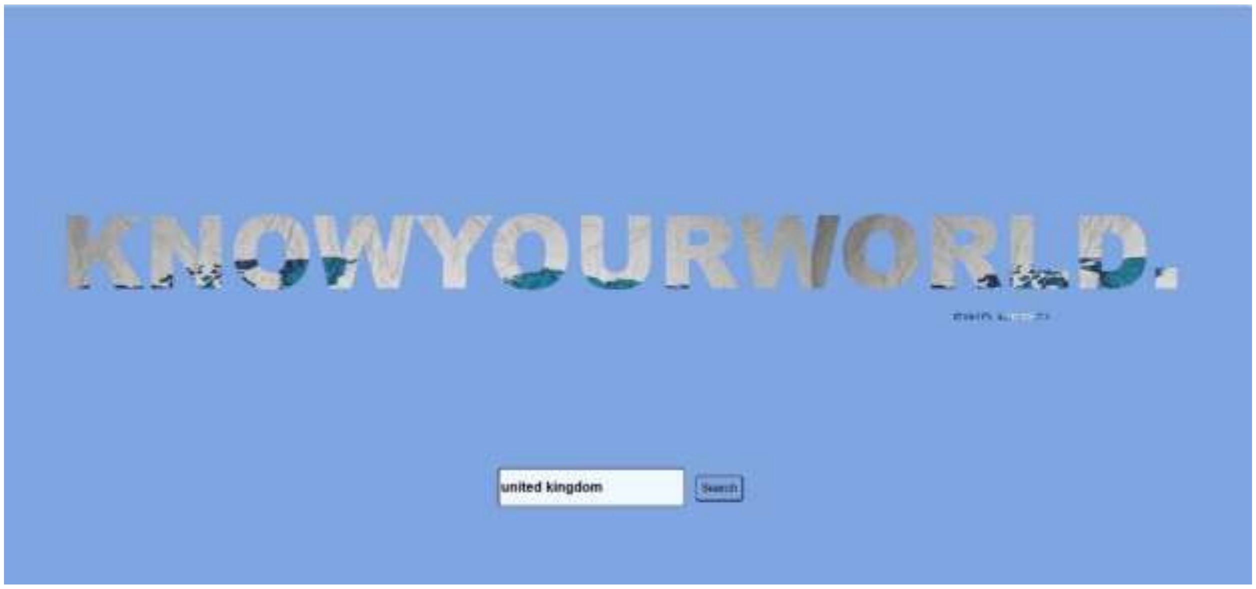
1. **Project Setup:**
   - **Directory Structure:**
     - ➢ Created a directory named called “knowYourWorld” to contain the project file.
     - ➢ Inside a directory, created separate files for HTML, CSS and JavaScript.

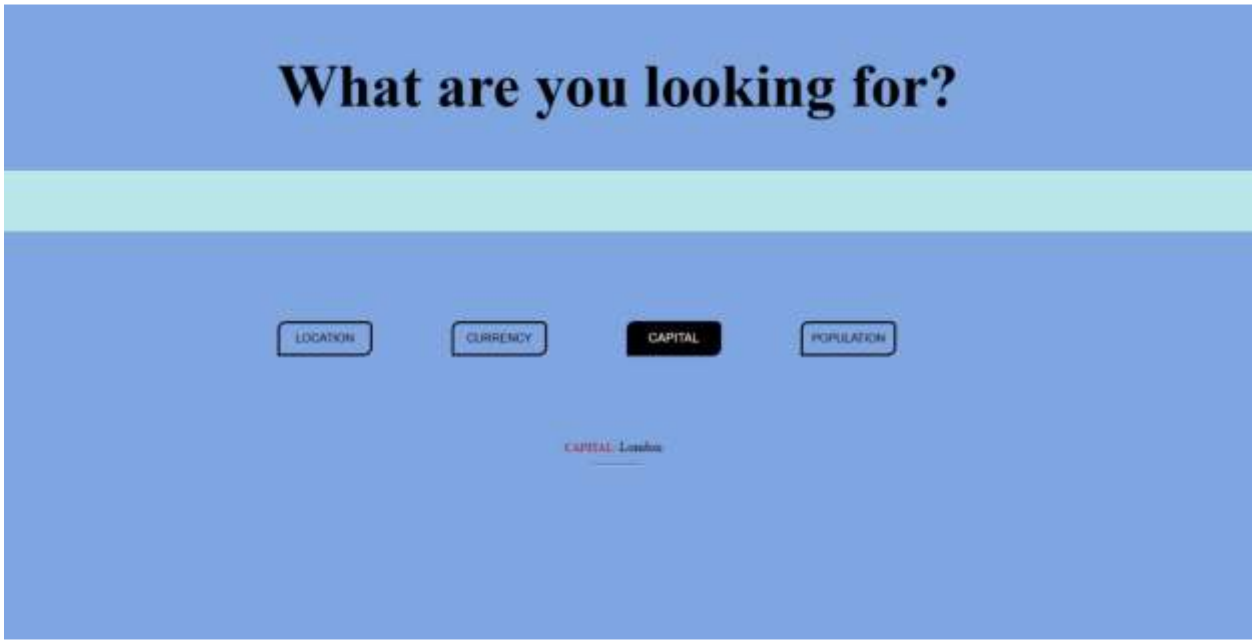
2. **Implementation Details:**
   - **HTML Structure:**
     - ➢ The HTML files defines the structure of the webpage, including an input field, button, and data display container, to ensure a visually appealing and user-friendly interface.
     - ➢ It also includes a container ‘<div>’ for displaying the fetched data by clicking on the button.

   - **CSS Styling:**
     - ➢ The CSS file provides styling for various elements, including the input field, buttons and data display container, to ensure visually appealing and user-friendly interface.

   - **JavaScript Logic:**
     - ➢ The JS file contains the logic of the project.
     - ➢ It listens for the click button on the search button and retrieves the entered country name from the input field.
     - ➢ It then makes the fetch request to the API endpoints to retrieve country position data based on the entered country name.
     - ➢ Upon receiving the response, it parses the data and dynamically updates the webpage to display the latitude and longitude of the country.
     - ➢ Error handling is implemented to display appropriate message in case of API errors or invalid input.

# Chapter 5: Working Project





CAPITAL: London