Project Guidelines and Requirements

For your final project, you will be creating an interactive portfolio (or any other project of your choice) that demonstrates your skills in jQuery. Below are the **minimum requirements** for your project. These are designed to help guide your implementation, but you are encouraged to get creative and go beyond these basics. If you'd prefer to work on a project other than a portfolio, feel free to do so, as long as you meet **minimum requirements** listed below.

Minimum Requirements:

1. **Theme and Styling**
    - Implement a **theme** (e.g., toggle between light and dark mode).
2. **Interactive effects and animations**
   Use **at least three** JavaScript/jQuery-based visual effects, such as:
    - Fade-in or slide-up effects when scrolling.
    - Smooth scrolling for navigation.
    - Hover animations (e.g., buttons changing color/size).
    - Elements appearing dynamically when the user interacts.
3. **At least 1 jQuery Plugin**
   Use **at least one jQuery plugin** to enhance the functionality of your project. Some examples:
    - **Carousel** for rotating images or testimonials.
    - **Lightbox** to display images in a modal.
    - **Sticky navigation bar** that stays at the top as users scroll.
    - **Tooltip** to display additional information on hover.
4. **At least 2 jQuery UI Widgets**
   Your project **must** include at least two jQuery UI widgets. Examples include:
    - Accordion (collapsible sections)
    - Tabs
    - Dialog (modal windows)
    - Datepicker
    - Slider
5. **Implement atleast 1 jQuery UI Interactions**
    - Draggable, resizable, sortable
6. **Form Validation**
   Implement **form validation** using the **jQuery Validation Plugin**. Your project should include a contact form and validate them to ensure the data is correct before submission.
7. **One API Integration Using AJAX**
   Fetch data from an external API using **AJAX**. You must dynamically load and display data on the page. Examples of APIs you can use include:
    - **Weather API** (e.g., OpenWeather API)
    - **GitHub API** (to display your repositories or projects)
    - **Quotes API** (to show random motivational quotes)

Ideas to Implement:

Here are some ideas you can consider:

- **Interactive Content & Data Display**
  - Implement an **Accordion** or collapsible sections to showcase projects, work experience, or education.
  - Add **image hover effects** (e.g., zoom-in, opacity change) for your portfolio images.
  - **Fun features: Interactive 'About Me' Page with Fun Facts**: Add an interactive section where users can click or hover to reveal fun facts about you, such as hobbies, favorite books, or future goals. Use jQuery or CSS animations to reveal the information in an engaging and interactive manner.
  - **Personalized Time-based Greeting**: Greet users based on the time of day (e.g., "Good Morning", "Good Afternoon"). Use JavaScript's `Date` object to dynamically generate a greeting message when the page loads.
- **Fetching Data from APIs**
  - **Weather Data**: Use an API like OpenWeather to display real-time weather data.
  - **GitHub Repositories**: Fetch and display your GitHub projects dynamically.
  - **Random Quotes**: Display dynamic quotes or testimonials from an API.
- **Advanced Interactivity**
  - Implement a **customizable theme** with an option for users to switch between light and dark modes.
  - Embed a **Google Map API** to show locations related to your projects or work.
- **Animations & Effects**
  - Add **smooth scrolling** between sections of your page.
  - Use **animated modals** to show more detailed project information when clicked.
  - Add a **testimonials section** with a carousel that rotates through quotes or reviews from colleagues, professors, or mentors. You can use jQuery UI's **carousel** widget or create a custom carousel to rotate the testimonials.

While the guidelines provide minimum requirements, **feel free to add more features**, implement extra functionality, and push the limits of what you can do. Just remember, **whatever project you choose**, the **minimum requirements** must still be met.

Reference Report:

Personal Portfolio Website

1. Title Page

- **Course Name**
- **Project Title**
- **Team Members with student ID**:

2. Table of Contents

On new page,

1. Introduction

Project Overview

The Portfolio Website serves as a personal, interactive online platform showcasing my skills, projects, and achievements as a web developer. Designed with an emphasis on interactivity, the website demonstrates various dynamic features such as smooth scrolling, real-time API data fetching, and dynamic content animations (like fade-ins and slide-ups). The aim of this project is to create a user-friendly, engaging website that not only highlights my technical abilities but also provides a smooth and enjoyable user experience.

Problem Statement

In the current digital age, having an online presence is essential for web developers to present their work. The challenge was to design a portfolio website that does more than simply display content. It needed to stand out with interactive elements, seamless transitions, and dynamic features, showcasing my development skills while offering a clean, professional interface. The website integrates jQuery and jQuery UI elements, API data fetching, and engaging visual effects to present my projects, experience, and skills in a way that attracts and retains visitors.

Technologies Used

The website was built using a range of technologies to ensure a responsive, interactive experience:

- **HTML5**: Provides the structure of the website's content.
- **CSS3**: Used for styling, responsive design, and ensuring mobile-first usability.
- **JavaScript/jQuery**: Implemented for dynamic updates and interactivity (e.g., animations, smooth scrolling, hover effects).
- **jQuery UI**: Used for UI widgets such as accordions and modals.
- **AJAX**: Used to dynamically fetch data from external APIs, such as weather information and GitHub repositories.
- **Bootstrap 4**: Provides the foundation for the layout, ensuring responsive design.

2. Features and Functionality

Main Features

1. **Accordion for Projects**:
    - The portfolio website includes an accordion feature that allows users to expand and collapse different sections of their projects. This helps in organizing the content in a way that's both compact and easily navigable. Each section includes detailed descriptions of the project, technologies used, and outcomes.
2. **Contact Form with Validation**:
    - A contact form was added to allow visitors to reach out to the portfolio owner. This form includes fields for name, email, and message, and features validation to ensure that users fill out the form correctly before submitting.
3. **Dynamic Data Fetching**:
    - Real-time data is fetched from external APIs. For instance, the weather API is used to fetch weather information and display it to the user based on their location. The GitHub API fetches a list of repositories, showcasing the developer's projects.
4. **Hover Effects**:
    - Hover effects are added to images to make the portfolio visually appealing. When the user hovers over project images, the opacity changes, or additional details appear, making the website more interactive.
5. **Smooth Scrolling**:
    - The website features smooth scrolling navigation, allowing users to navigate from one section to another seamlessly. This enhances the user experience by providing smooth transitions between sections when clicking on the navigation menu.

3. Implementation Details

The project was developed with a focus on usability and functionality. Below are some important code snippets and their functionality:

Accordion Widget

The accordion feature was created using jQuery UI's accordion widget, which allows users to expand and collapse different sections.

- **Place a code snippet**
- **Purpose**: The accordion collapses sections by default and expands when the user clicks on them, showing detailed project descriptions.

Form Validation

The contact form uses the jQuery Validation Plugin to ensure that all required fields are filled out correctly before submission.

- **Place a code snippet**
- **Purpose**: Ensures that the contact form is properly validated before submission, preventing errors in the communication process.

AJAX API for Weather

AJAX was used to make an asynchronous request to the OpenWeather API, dynamically fetching the current weather.

- **Place a code snippet**
- **Purpose**: Fetches real-time weather data and updates the website with the current temperature for a specified location.

4 Conclusion

5. References

- **OpenWeather API**: https://openweathermap.org/api
- **GitHub API**: https://developer.github.com/v3/
- **jQuery**: https://jquery.com/
- **jQuery UI**: https://jqueryui.com/
- **Bootstrap 4**: https://getbootstrap.com/

Formatting Guidelines

- **Font**: Times New Roman, 12 pt.
- **Spacing**: 1.5 or double line spacing.
- **Margins**: Standard 1-inch margins on all sides.
- **Page Numbers**: Include page numbers in the footer.
- **Headings**: Use consistent heading styles (e.g., Heading 1 for major sections, Heading 2 for subsections).