Front End Engineering-II

Project Report

Semester-IV (Batch-2022)

Title of the Project: TodoList

A red and white sign

Description automatically generated with low confidence

**Supervised By: Submitted By:**

Raveesh Samkaria Name:Suraj Kumar

Roll no: 2210990877

Group: G-13

**Github Profile Link:https://github.com/surajkumar752894/Todolist**

**Department of Computer Science and Engineering**

## Chitkara University Institute of Engineering & Technology,

## Chitkara University, Punjab

**CONTENTS:**

|  |  |  |
| --- | --- | --- |
| **S.NO** | **CONTENT** | **PAGE.NO** |
| **1** | **Abstract** | **1** |
| **2** | **Introduction** | **2** |
| **3** | **Problem Definition and Requirements** | **3** |
| **4** | **Proposed design** | **4-6** |
| **5** | **HTML code** | **6-7** |
| **6** | **CSS code** | **7-8** |
| **7** | **Js code** | **8-9** |
| **8** | **Result** | **9-11** |

**Abstract:**

HTML (Hypertext Markup Language) and CSS (Cascading Style Sheets) form the backbone of web development.

**HTML (Hypertext Markup Language)** is the standard markup language for creating web pages and web applications. Let’s break down what HTML means:

**Hyper Text**: Hyper-text refers to “text within text.” When a piece of text contains a link, it becomes a hypertext. Clicking on a link takes you to a new webpage, and that action is considered hypertext. Essentially, hypertext allows you to link different web pages (which are HTML documents) together.

**Markup Language**: A markup language is a computer language used to apply layout and formatting conventions to a text document. It makes text more interactive and dynamic. With HTML, you can turn plain text into images, tables, links, and more.

**Web Page**: A web page is a document commonly written in HTML and interpreted by a web browser. You can identify a web page by entering its URL. Web pages can be static or dynamic. HTML allows us to create static web pages that display nicely in web browsers.

**Let’s break down the components:**

**<!DOCTYPE>:** Defines the document type or instructs the browser about the HTML version.

**<html>:** Informs the browser that it’s an HTML document. The content between the <html> tags describes the web document.

**<head>:** Contains metadata (information about the document) and should be the first element inside <html>.

**<title>**: Adds the title of the HTML page (appears at the top of the browser window).

**<body>:** Contains the main content visible to the end user.

**CSS (Cascading Style Sheets)** is a fundamental technology used in web development. Let’s dive into what CSS entails:

**Definition**:

**CSS** is a **style sheet language** that specifies how HTML elements should be **presented and styled** on various media, including screens, paper, and other devices.

It works alongside HTML and JavaScript to create visually appealing and well-organized web pages.

**Key Points**:

**Presentation and Styling**: CSS defines the visual appearance of web content. It controls aspects such as fonts, colors, spacing, layout, and positioning.

**Cascading**: The term “cascading” refers to the order in which styles are applied. Multiple styles can be combined, and the most specific rule takes precedence.

**Stylesheets**: CSS rules are typically stored in external .css files. These files contain style definitions that can be reused across multiple web pages.

**Efficiency**: By using CSS, you can maintain a consistent look and feel across an entire website without duplicating code.

**JavaScript (**is a high-level, interpreted programming language primarily used for building dynamic and interactive web pages. Here's a brief introduction to JavaScript:**)**

1. **\*\*Purpose\*\*:** JavaScript was initially created to add interactivity and dynamic behavior to web pages. It allows developers to manipulate the content of web pages, respond to user actions, and create interactive web applications.

2. **\*\*Client-Side Scripting\*\***: JavaScript is mainly executed on the client-side, meaning it runs in the user's web browser. This enables developers to create responsive and interactive user interfaces without relying on server-side processing.

3. **\*\*Syntax\*\*:** JavaScript syntax is similar to other programming languages like Java and C, making it relatively easy to learn for developers familiar with those languages. It's a dynamically typed language, meaning variable types are determined at runtime.

**4. \*\*Features\*\*:**

- \*\*Variables and Data Types\*\*: JavaScript supports various data types such as numbers, strings, booleans, objects, arrays, and functions.

**- \*\*Functions\*\*:** Functions in JavaScript are first-class citizens, meaning they can be assigned to variables, passed as arguments, and returned from other functions.

- \*\*DOM Manipulation\*\*: JavaScript allows manipulation of the Document Object Model (DOM) which represents the structure of HTML documents. This enables developers to dynamically change the content, structure, and style of web pages.

- \*\*Event Handling\*\*: JavaScript provides mechanisms for handling user interactions, such as clicks, mouse movements, and keyboard inputs, through event listeners.

- \*\*Asynchronous Programming\*\*: JavaScript supports asynchronous programming using callbacks, promises, and async/await syntax, allowing non-blocking execution of code, especially when dealing with tasks like fetching data from servers or handling user input.

5. **\*\*Frameworks and Libraries\*\*:** JavaScript has a rich ecosystem of frameworks and libraries that extend its capabilities and simplify common tasks. Popular libraries include jQuery for DOM manipulation, React.js for building user interfaces, AngularJS for web app development, and Express.js for server-side JavaScript.

6. **\*\*Compatibility\*\***: JavaScript is supported by all major web browsers, including Chrome, Firefox, Safari, and Edge, making it a versatile choice for web development.

Overall, JavaScript is a powerful language that empowers developers to create dynamic, interactive, and feature-rich web applications, ranging from simple web pages to complex single-page applications (SPAs) and progressive web apps (PWAs). Its flexibility, ease of use, and wide adoption make it an essential tool for web development.

**Introduction:**

**Background:**Provide context for the project, explaining why it is relevant and necessary.

**Objectives:**Clearly state the goals and objectives of the project.

**Significance:**Discuss the importance and potential impact of the project.

**Problem Definition and Requirements**

**Problem Statement:**Describe the specific problem or challenge addressed by the project.

**Software Requirements:**List the software tools, libraries, and technologies used.

****Text Editors / Integrated Development Environments (IDEs)**:**

**Visual Studio Code (VS Code)**: A popular free code editor with excellent HTML and CSS support.

**Browser Extensions**:

**Live Server**: Instantly reloads your HTML and CSS changes.

**CSS Viewer**: Inspect CSS properties on any webpage.

**Hardware Requirements:**Specify any hardware components required for the project.

**Web Browser**:

Install popular web browsers for testing your HTML and CSS:

**Google Chrome**

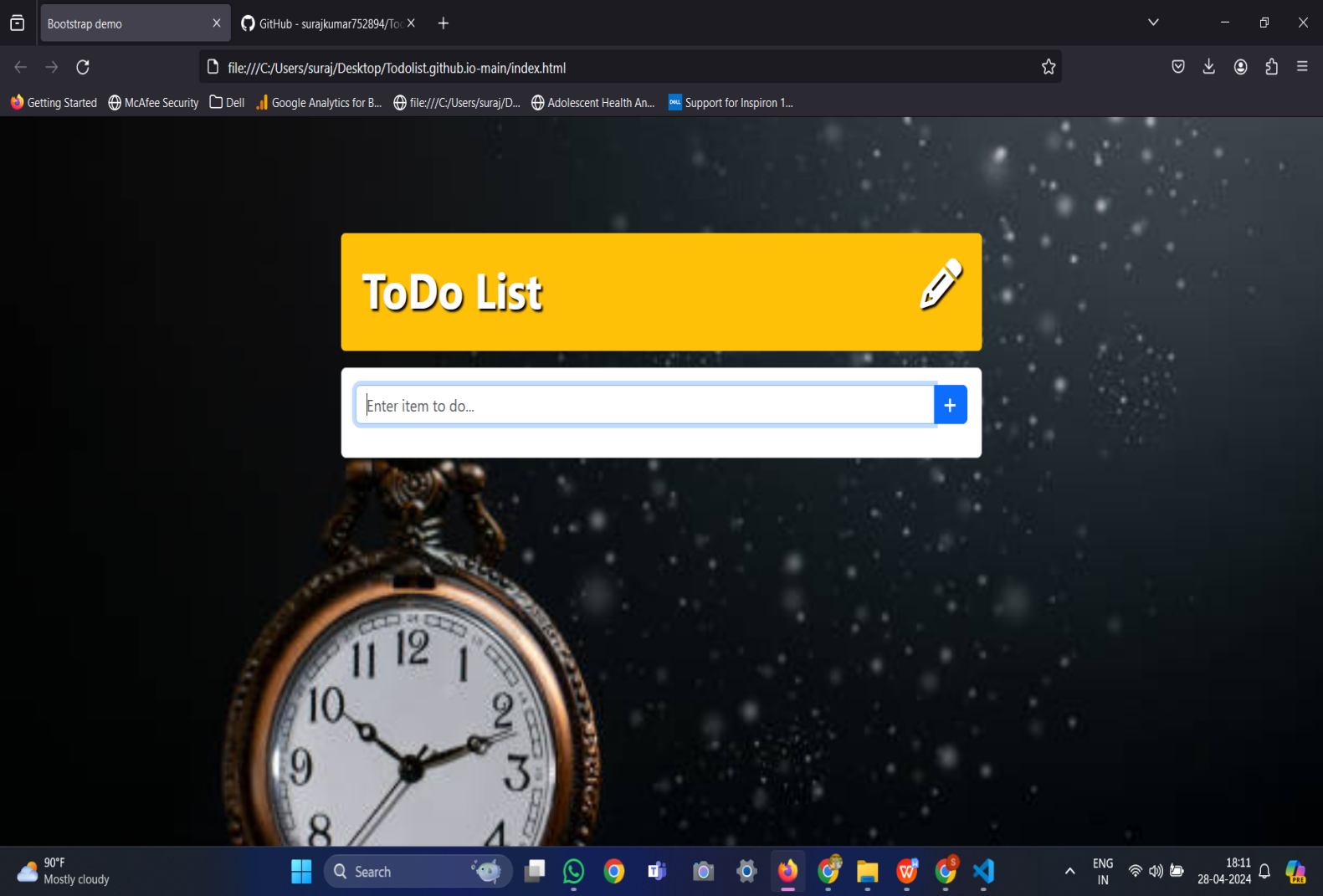
**Text Editor or IDE**:

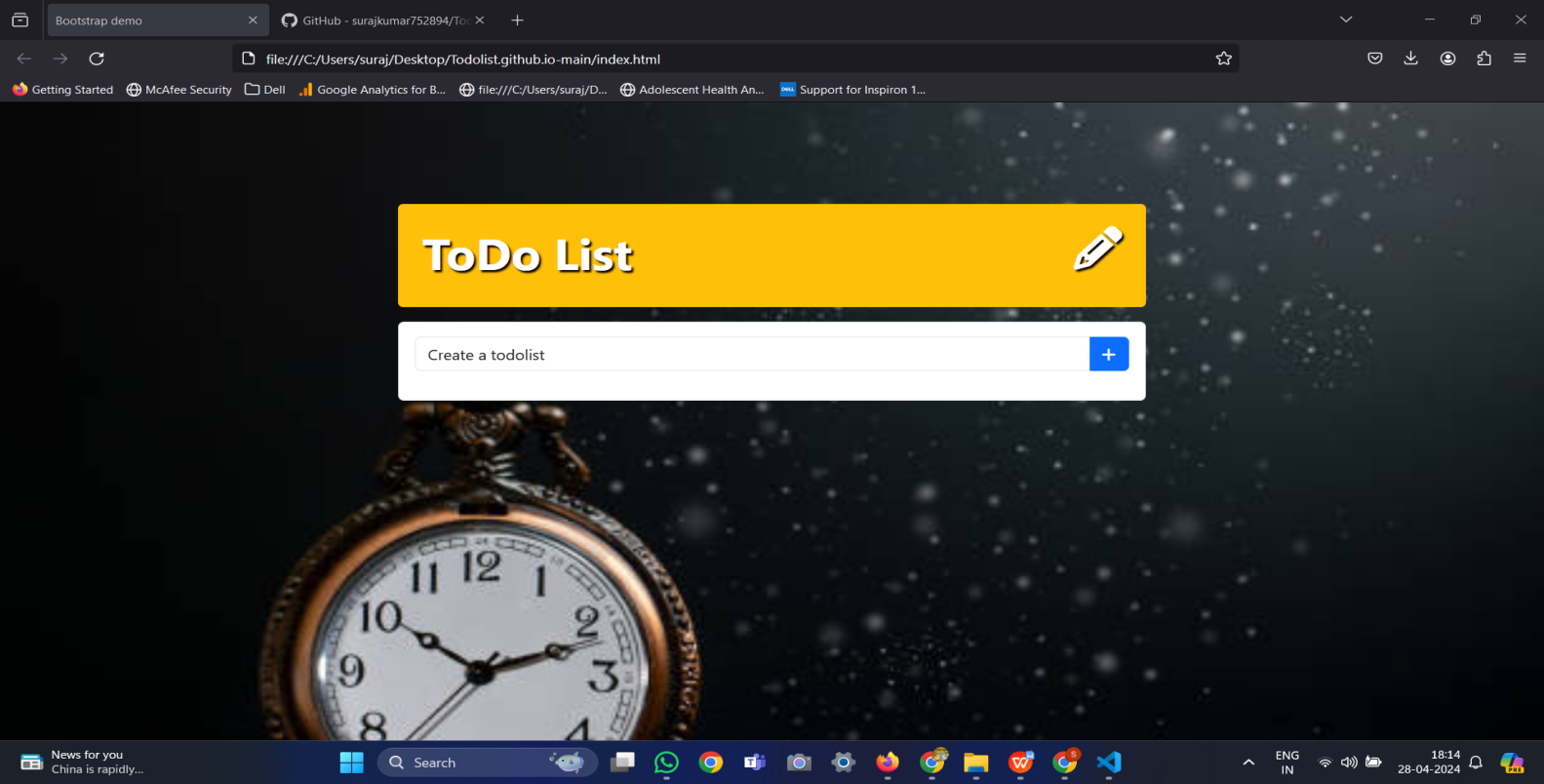
Choose a text editor or integrated development environment (IDE) for writing code:

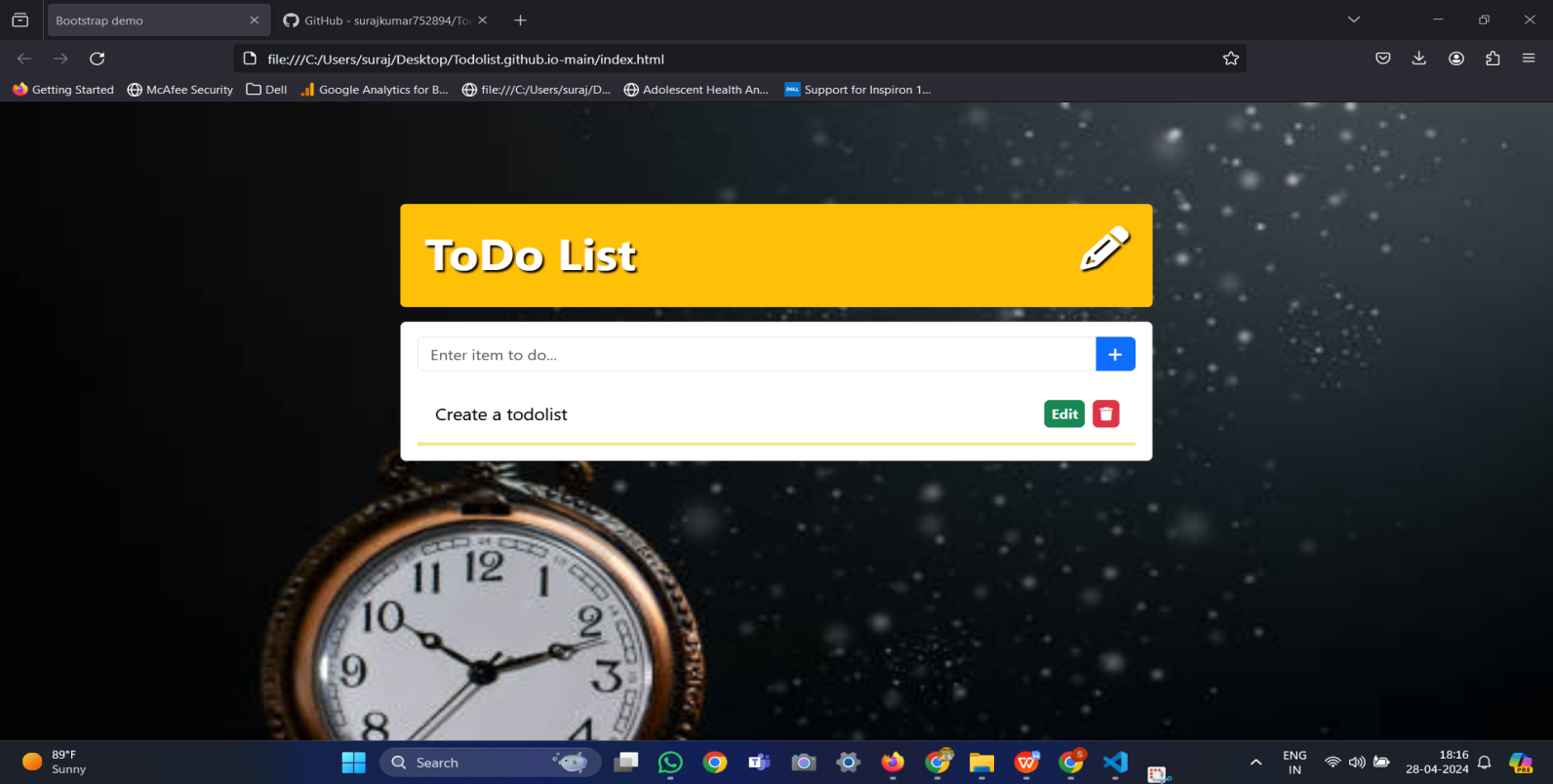
**Visual Studio Code (VS Code)**

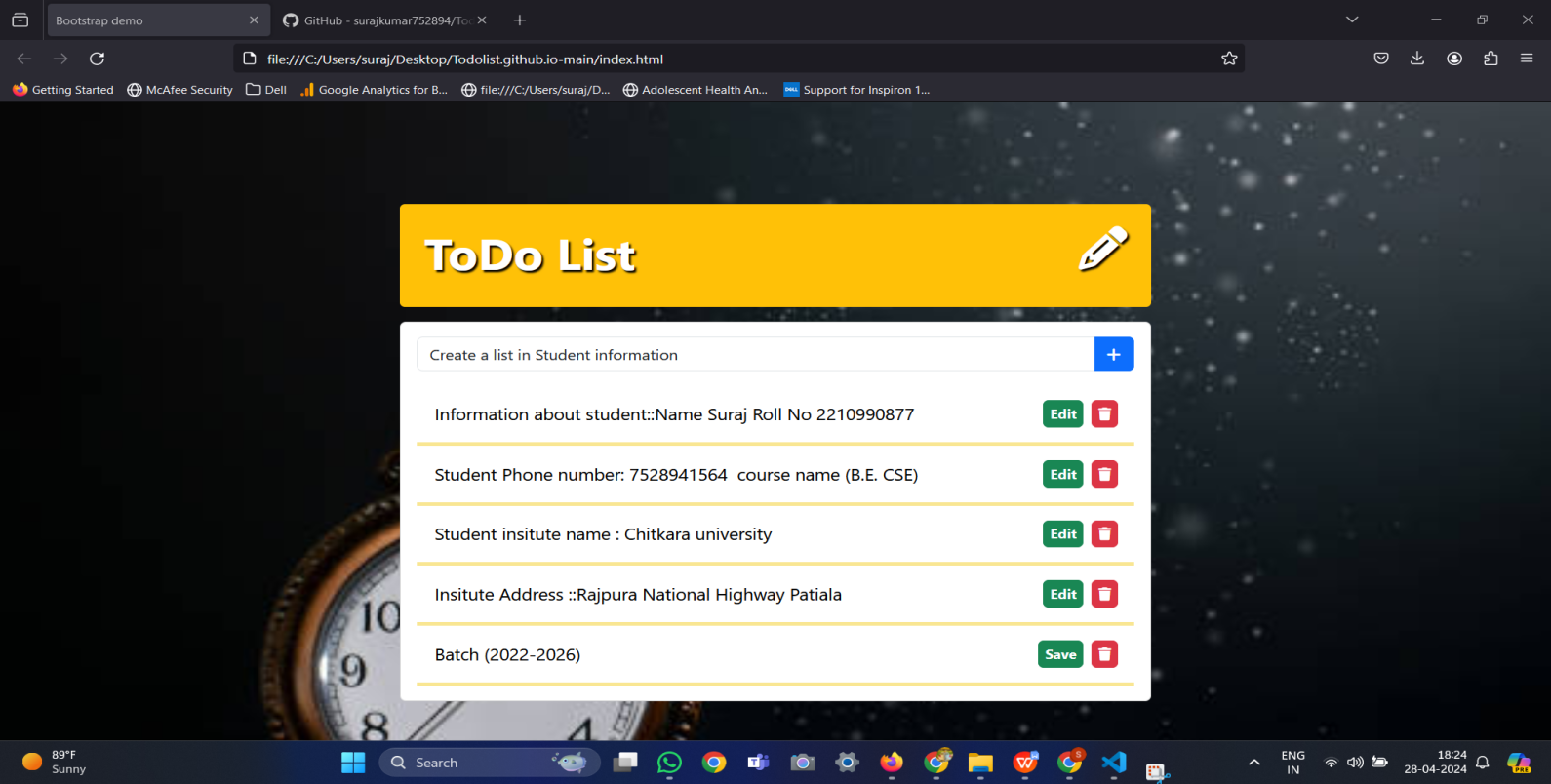
**Proposed Design / Methodology:**

**Design:**



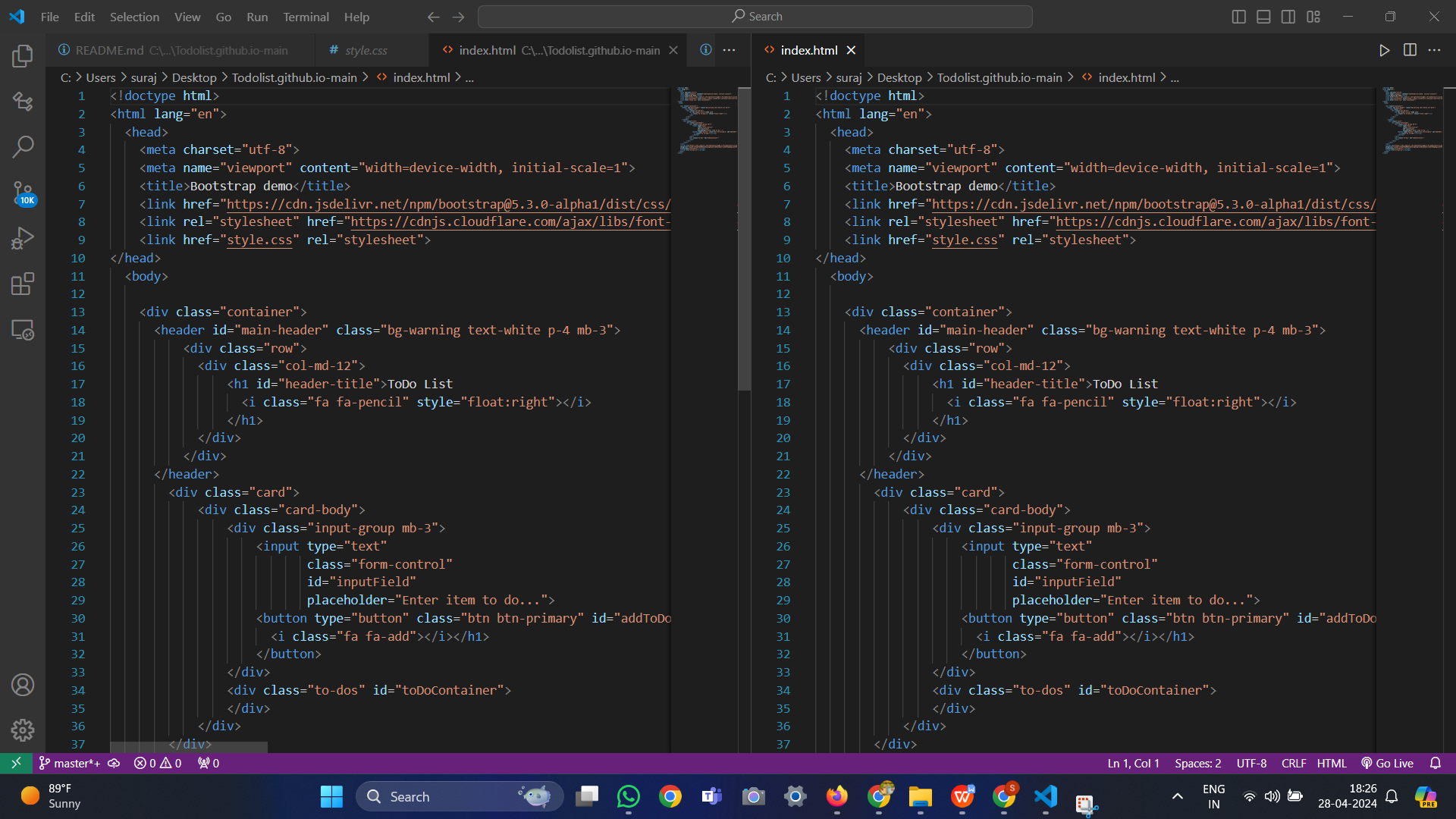


****

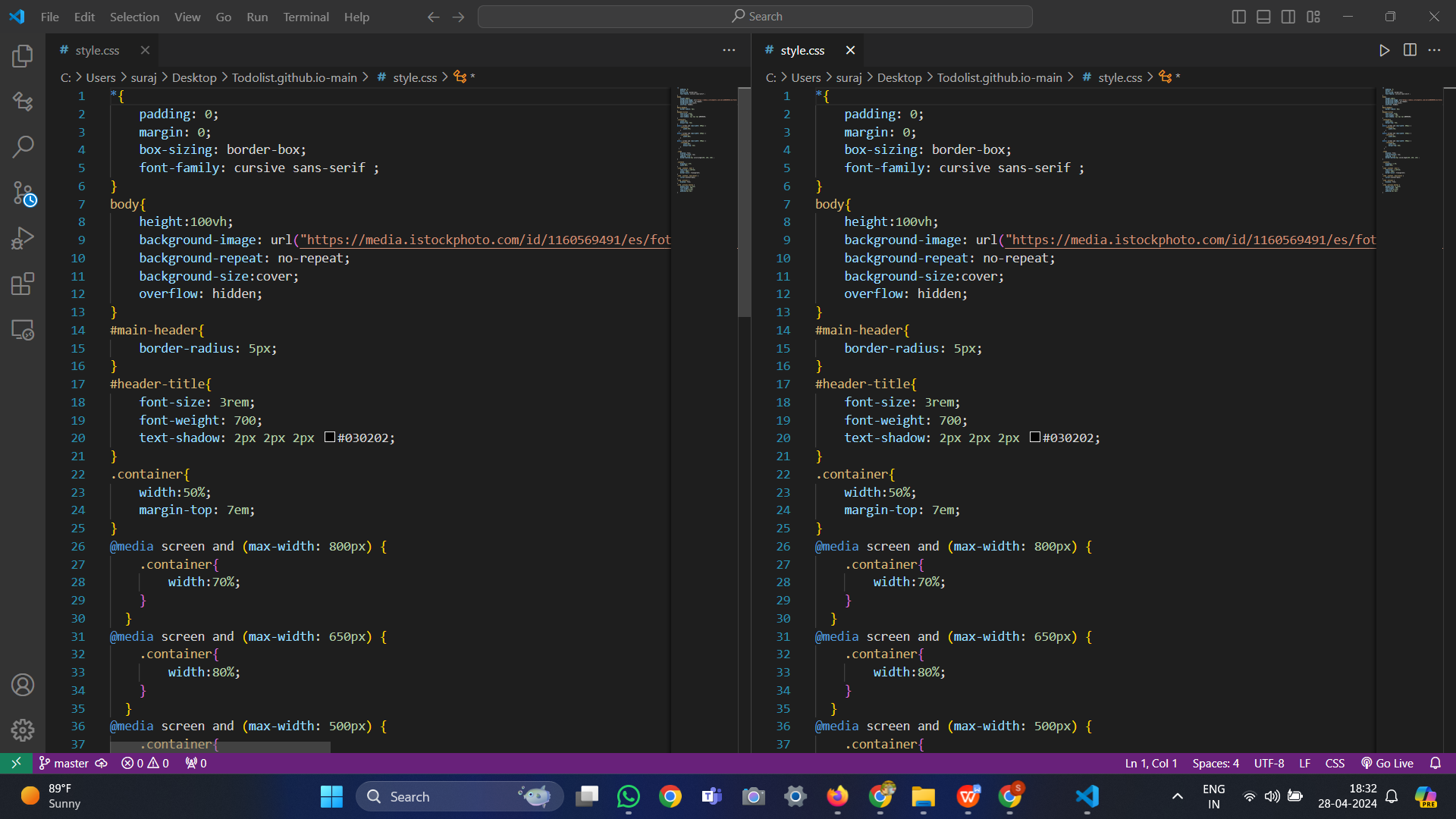
****

**Source code:**

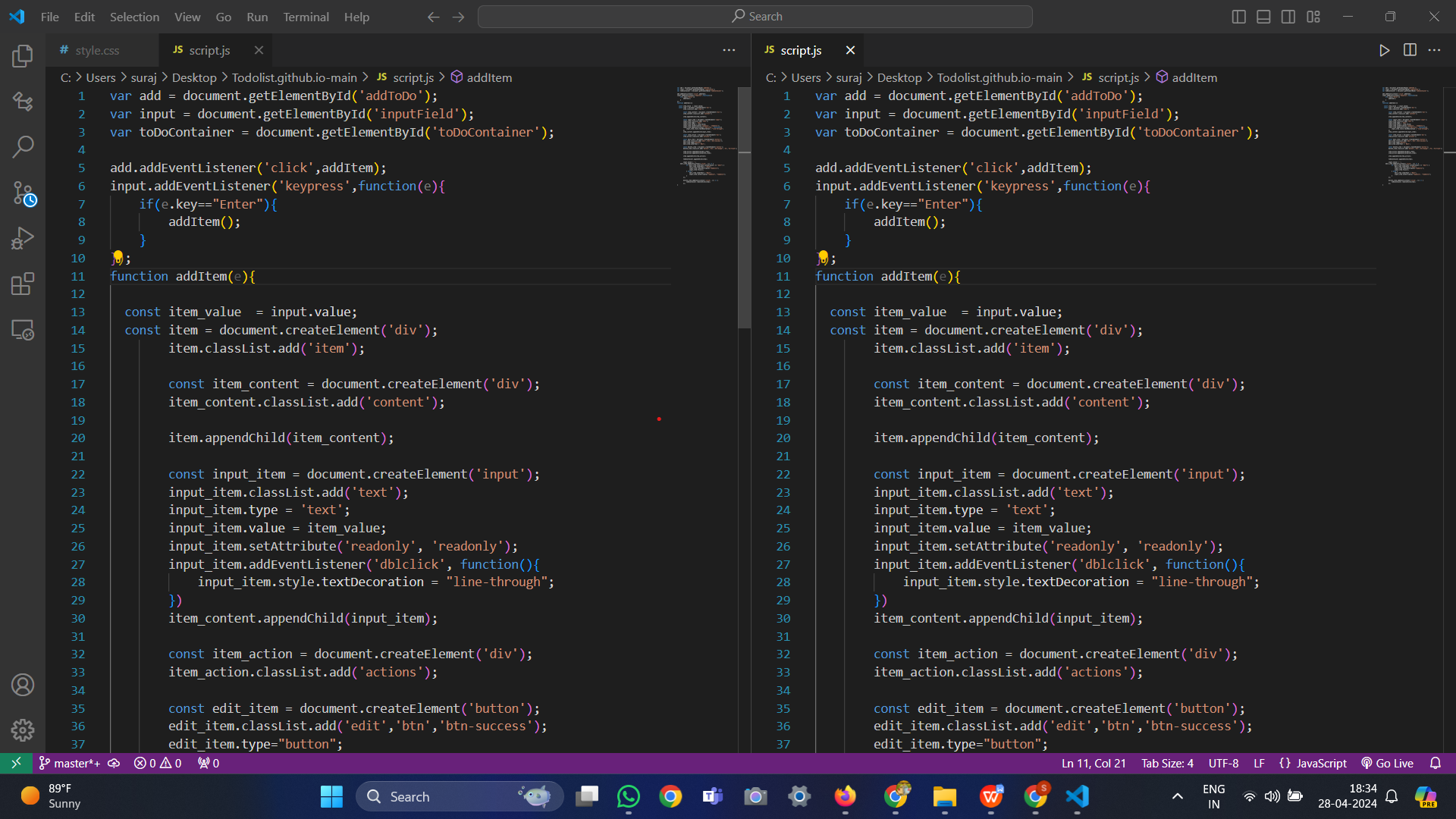
**HTML CODE:**



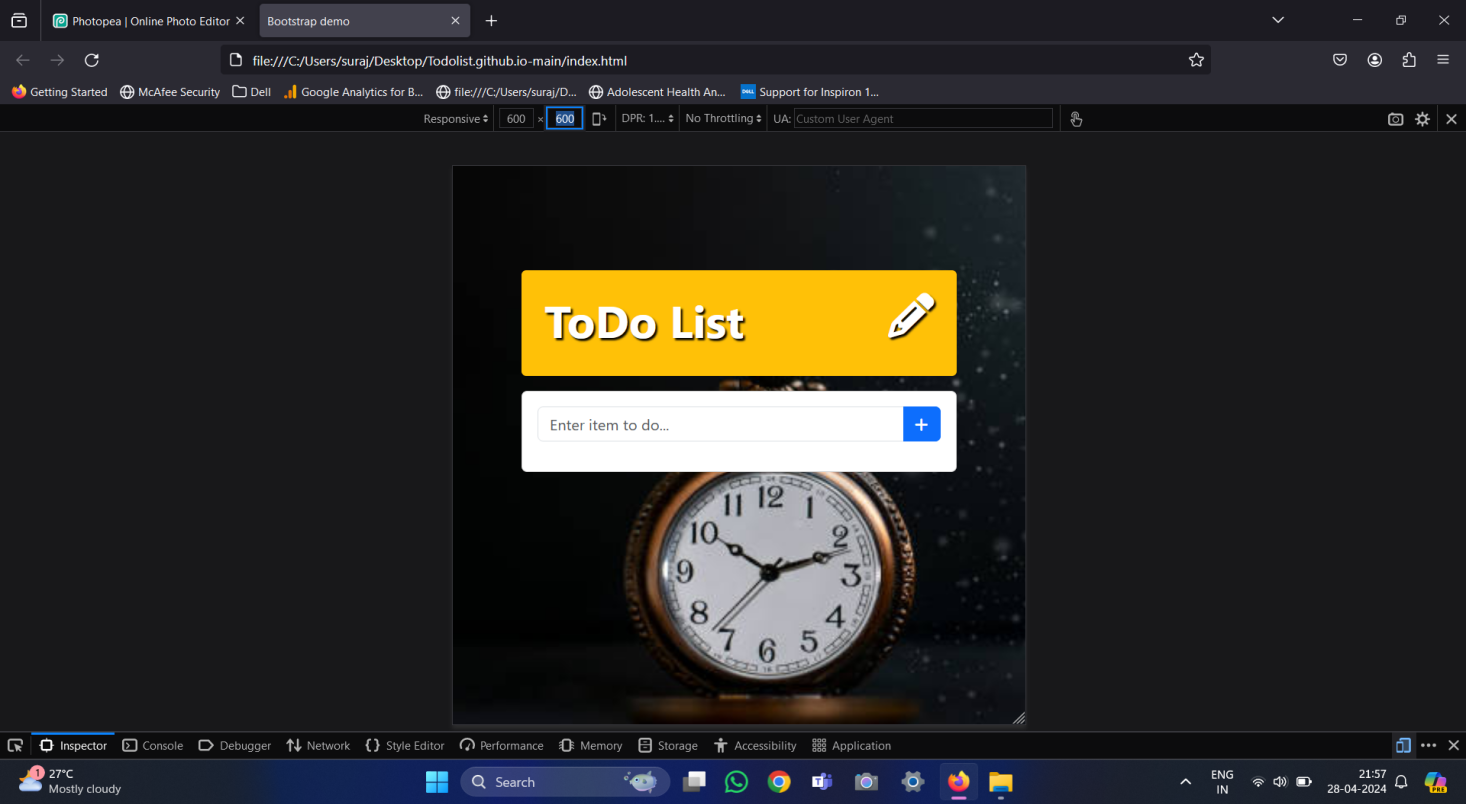
**CSS CODE: DESKTOPE**

****

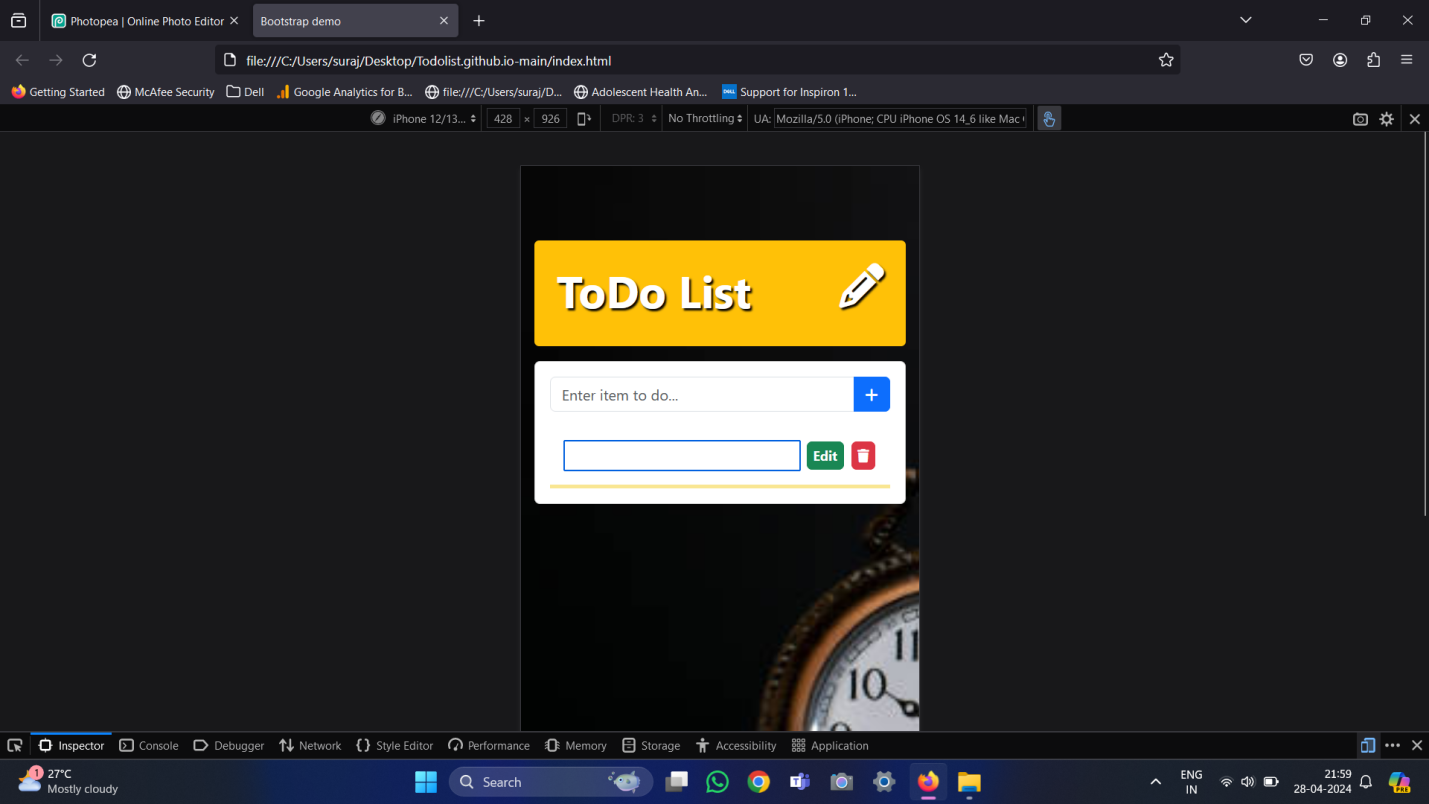
**JAVASCRIPT CODE: DESKTOP**

****

**For Max Width::600Px;**

****

**PHONE View::**



Ipad view::

