“WebDojo DPS”

A Minor Project Report

Submitted in Partial Fulfillment of requirements for the Award of Degree of Bachelor of Engineering in Computer Science & Engineering
Submitted to

RAJIV GANDHI PROUDYOGIKI VISHWAVIDYALAYA, BHOPAL (M.P)

Submitted By
Pranjal Kumar Dwivedi
0132CS211115

Under the guidance of
POOJA MEENA

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
RADHA RAMAN INSTITUTE OF TECHNOLOGY & SCIENCE, BHOPAL (M.P.)

Session: March - June 2024

RADHARAMAN INSTITUTE OF TECHNOLOGY & SCIENCE
BHOPAL (M.P.)

Department of Computer Science & Engineering

APPROVAL CERTIFICATE

This Minor project work entitled **“WebDojo DPS”** being submitted by **“PRANJAL KUMAR DWIVEDI” 0132CS211115** is/are approved for the award of degree of **Bachelor of Engineering** in **Computer Science and Engineering**.

Internal Examiner **External Examiner**

Date: **Date:**

RADHARAMAN INSTITUTE OF TECHNOLOGY & SCIENCE
BHOPAL (M.P.)

Department of Computer Science & Engineering

CANDIDATE DECLARATION

We/I **“PRANJAL KUMAR DWIVEDI ” (Roll No.: 0132CS211115)** students of **Bachelor of Engineering** in **Computer Science & Engineering, Radharaman Institute Of Technology & Science**, **Bhopal (M.P.),** hereby declare that the work presented in this Minor project “WebDojo DPS” is the outcome of our own work, is bonafide and correct to the best of our knowledge and this work has been carried out taking care of Engineering Ethics. The work presented does not infringe any patented work and has not been submitted to any other university or anywhere else for the award of any degree or any professional diploma.

Date:

“PRANJAL KUMAR DWIVEDI”
0132CS211115

ACKNOWLEDGEMENT

This project work is the result of guidance and support of various people at RITS without whom all our effort would have been directionless and fruitless. We sincerely thank all of them, for assisting us in completing the dissertation.

We express our ardent and earnest gratitude to our guide, **Prof. POOJA MEENA** Department of Computer Science & Engineering, RITS Bhopal and **Prof. Chetan Agrawal**, HOD, Department of Computer Science & Engineering, RITS Bhopal for their help and encouragement at all the stages of our Work. Their guidance and motivation helped us to be fruitful in our effort.

We also express my heartfelt and profound gratitude to our Director AJAY SINGH for his valuable suggestion and ample resources at all stages of the research work.

Finally, we would like to say that we are indebted to my parents for everything that they have done for us. All of this would have been impossible without their constant support. And we also thank to God for being kind to me and driving me through this journey.

“PRANJAL KUMAR DWIVEDI”
0132CS211115

INDEX

Minor Project Report : WebDojo DPS

Chapter 1: Introduction

Overview

WebDojo DPS is an innovative web-based Integrated Development Environment (IDE) designed to streamline the web development process. It provides users with a single-window interface for writing, editing, and previewing HTML, CSS, and JavaScript code in real-time.

Problem Statement

Traditional IDEs often require developers to switch between multiple windows or tabs to edit different types of code, leading to inefficiency and inconvenience. WebDojo DPS addresses this issue by offering a unified environment for coding and previewing, enhancing the developer experience.

Objective

The primary objective of WebDojo DPS is to simplify the web development workflow by providing a user-friendly interface that eliminates the need for switching between multiple tools or environments. It aims to improve productivity and facilitate rapid prototyping for developers.

Applications

WebDojo DPS has diverse applications, including:

- Educational purposes for students learning web development.
- Professional web developers for quick prototyping and testing.
- Enthusiasts seeking a simple and efficient coding environment for personal projects.

Chapter 2: Requirement Analysis

1. Functional Requirements

- **Code Editing:** Users should be able to write, edit, and manage HTML, CSS, and JavaScript code within the application.
- **Real-time Preview:** Changes made to the code should be reflected immediately in the output preview window.
- **Syntax Highlighting:** The application should provide syntax highlighting for HTML, CSS, and JavaScript code to improve readability.
- **Code Synchronization:** Changes made in one editor (HTML, CSS, or JavaScript) should automatically synchronize with the other editors and the preview window.
- **User Interface:** The interface should be intuitive and user-friendly, with separate sections for HTML, CSS, and JavaScript editing.

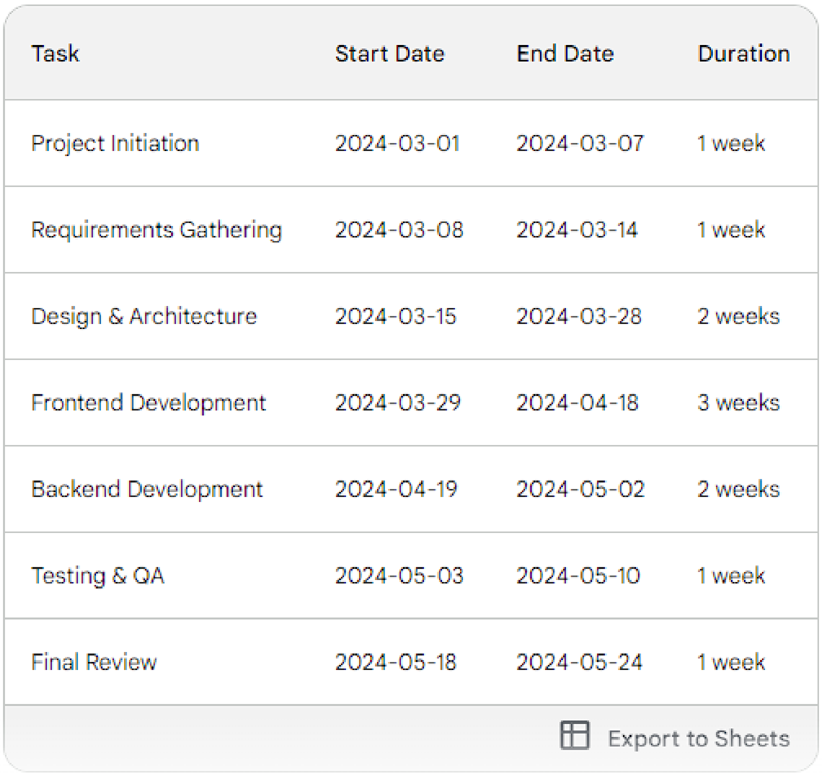
2. Non-Functional Requirements

- **Performance:** The application should be responsive and provide smooth real-time preview updates, even with large code bases.
- **Security:** WebDojo DPS should implement security measures to prevent unauthorized access and protect user data.
- **Compatibility:** The application should be compatible with modern web browsers to ensure a consistent user experience across platforms.
- **Usability:** The user interface should be designed with usability principles in mind, making it easy for users to write and preview code efficiently.
- **Scalability:** The system should be scalable to accommodate future enhancements and updates.

3. Constraints

- **Browser Compatibility:** The application must be compatible with major web browsers such as Chrome, Firefox, Safari, and Edge.
- **Resource Limitations:** The system should not consume excessive resources (CPU, memory) to ensure optimal performance on a wide range of devices.

Gantt Charts



Task	Start Date	End Date	Duration
Project Initiation	2024-03-01	2024-03-07	1 week
Requirements Gathering	2024-03-08	2024-03-14	1 week
Design & Architecture	2024-03-15	2024-03-28	2 weeks
Frontend Development	2024-03-29	2024-04-18	3 weeks
Backend Development	2024-04-19	2024-05-02	2 weeks
Testing & QA	2024-05-03	2024-05-10	1 week
Final Review	2024-05-18	2024-05-24	1 week

Chapter 3: Designing & Methodology

Overview of Methodology

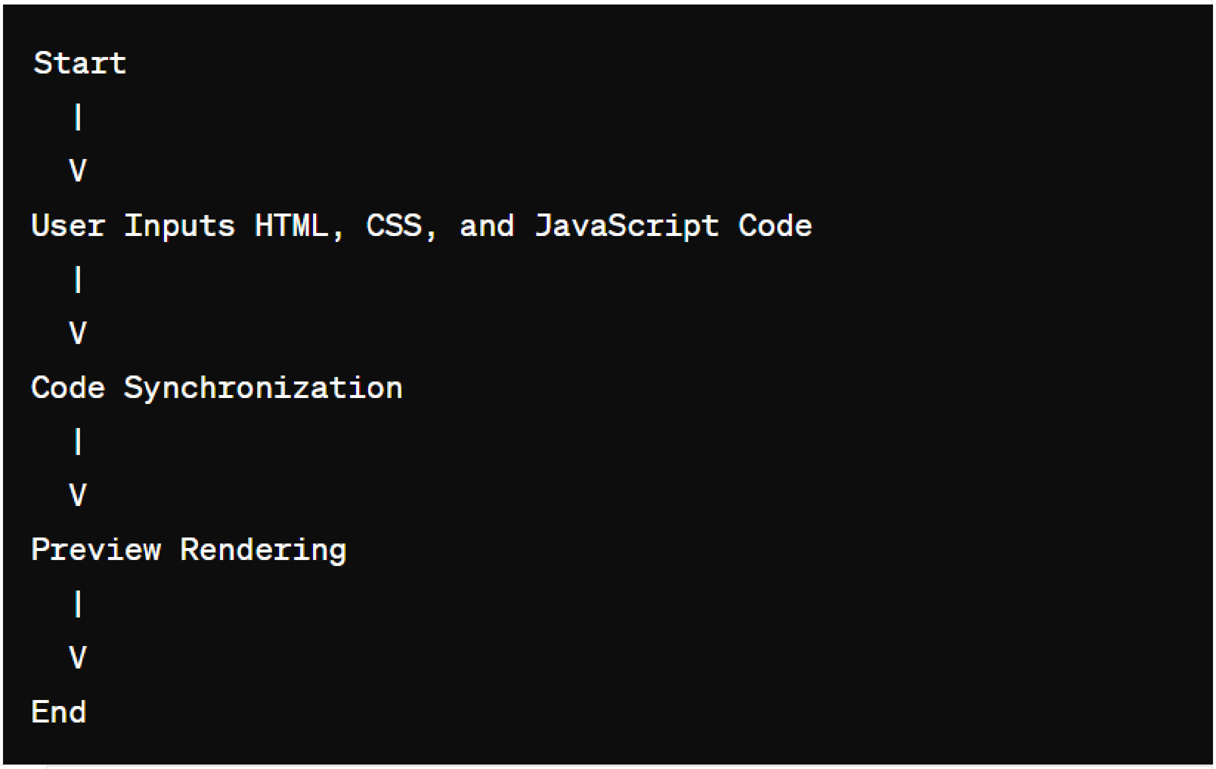
The development methodology for WebDojo DPS follows an agile approach, emphasizing iterative development and continuous feedback. It allows for flexibility and adaptation to changing requirements throughout the project lifecycle.

Detail Methods

WebDojo DPS utilizes HTML, CSS, and JavaScript for its user interface and functionality. It employs event-driven programming to synchronize code editing and real-time preview, ensuring a seamless user experience.

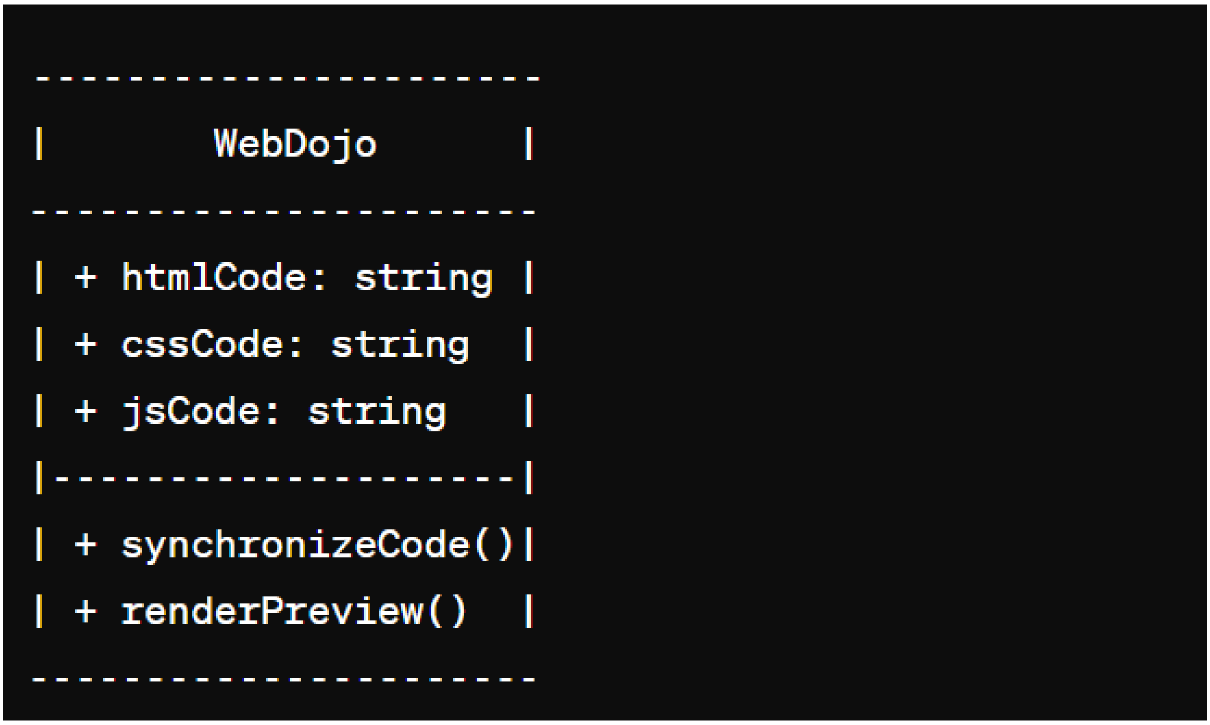
Diagrams:

Flow Chart:



```
Start
  |
  V
User Inputs HTML, CSS, and JavaScript Code
  |
  V
Code Synchronization
  |
  V
Preview Rendering
  |
  V
End
```

Class Diagram:--------



```
---------------------
|       WebDojo      |
---------------------
| + htmlCode: string |
| + cssCode: string  |
| + jsCode: string   |
|--------------------|
| + synchronizeCode()|
| + renderPreview()  |
---------------------
```

Algorithm

The algorithm implemented in WebDojo DPS manages the synchronization of code changes across HTML, CSS, and JavaScript editors, updating the preview window dynamically without page reloads.

Project Module

The project is modularized into components such as:

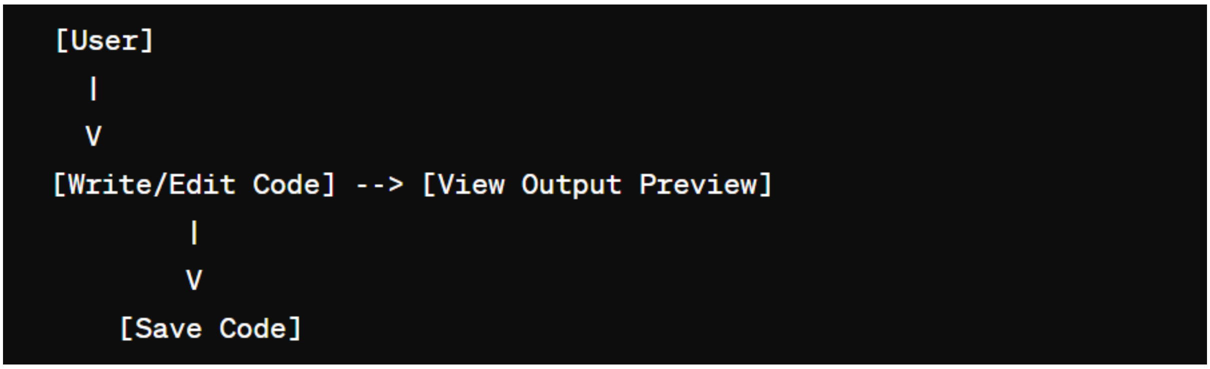
- User interface elements
- Code synchronization module
- Output preview window management

Architecture Diagram

An architecture diagram illustrates the high-level structure of WebDojo DPS, depicting the interactions between its components and external dependencies.

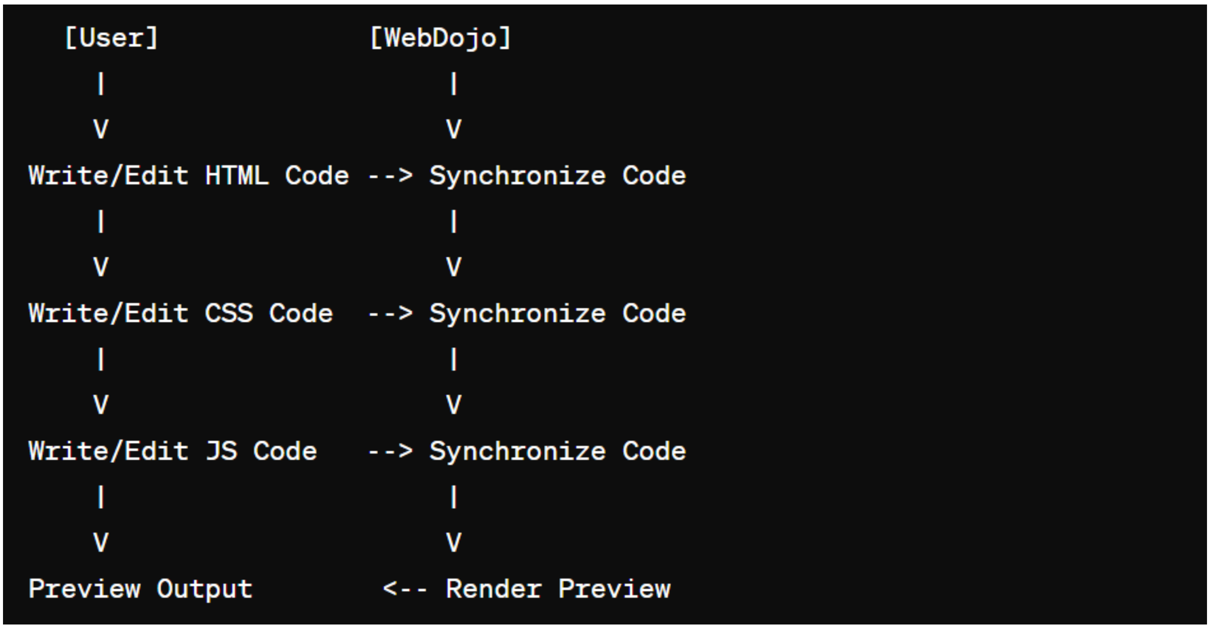
Diagrams

- **Use Case Diagram:**



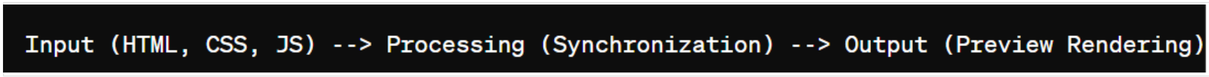
```
 [User]
   |
   V
[Write/Edit Code] --> [View Output Preview]
         |
         V
    [Save Code]
```

- **Sequence Diagram:**



```
  [User]               [WebDojo]
    |                      |
    V                      V
Write/Edit HTML Code --> Synchronize Code
    |                      |
    V                      V
Write/Edit CSS Code  --> Synchronize Code
    |                      |
    V                      V
Write/Edit JS Code   --> Synchronize Code
    |                      |
    V                      V
Preview Output        <-- Render Preview
```

DFD with Explanation:



```
Input (HTML, CSS, JS) --> Processing (Synchronization) --> Output (Preview Rendering)
```

Database Designing

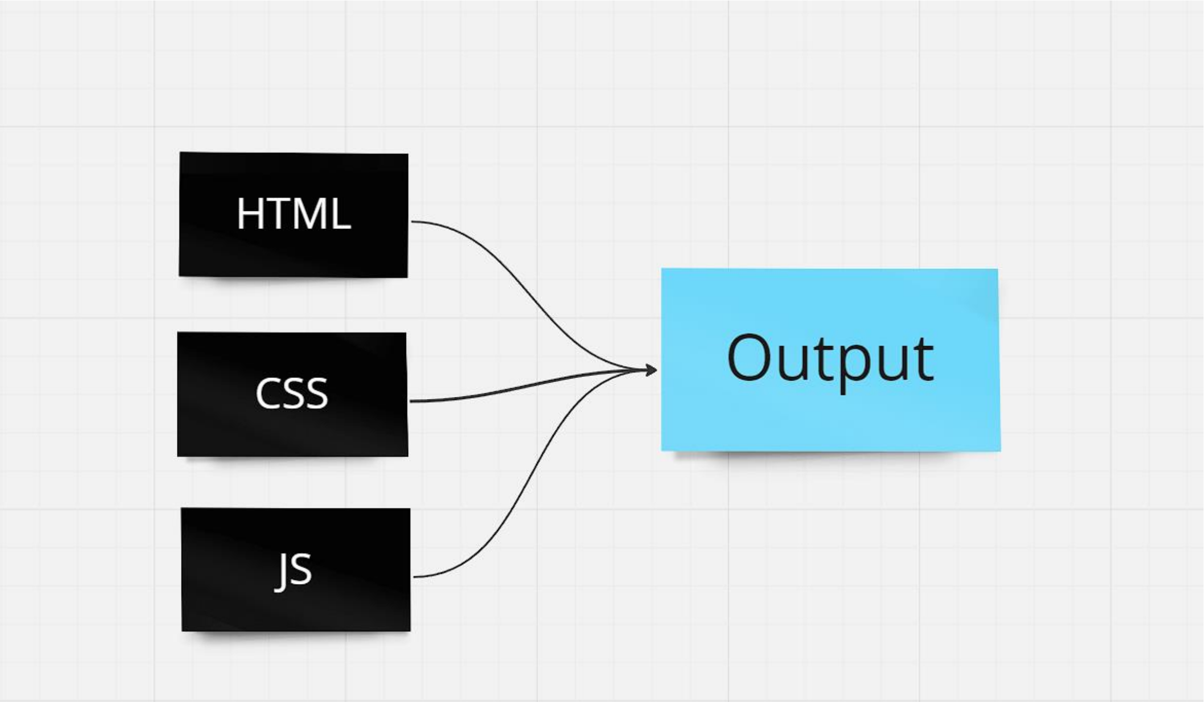
WebDojo DPS does not require a database as it primarily focuses on code editing and previewing without persistent data storage.

Data Dictionary

As there is no database involved, a data dictionary is not applicable for WebDojo DPS.

E-R Diagrams

Since there is no database component, Entity-Relationship diagrams are not relevant to WebDojo DPS.



Chapter 4: Implementation and Results

External Interface Requirements

User Interfaces

WebDojo DPS provides intuitive HTML, CSS, and JavaScript code editors, along with an output preview window for real-time feedback.

Hardware Interfaces

WebDojo DPS operates on standard hardware configurations and does not have specific hardware dependencies.

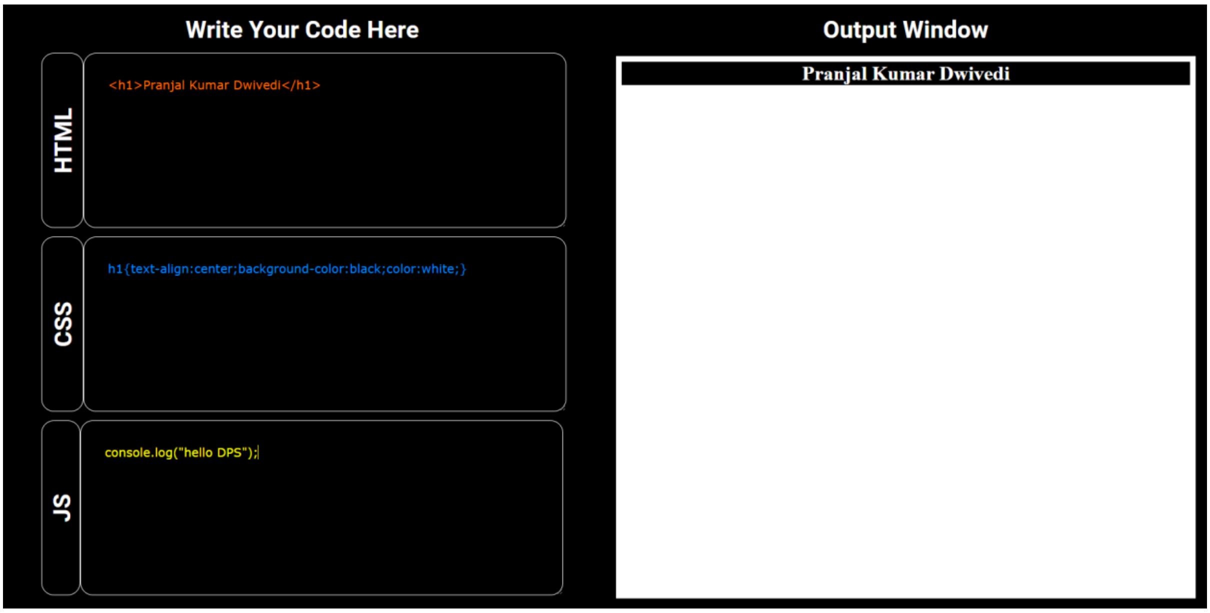
Software Interfaces

WebDojo DPS interacts with web browsers to render the user interface and execute JavaScript code. It utilizes GSAP (GreenSock Animation Platform) for animations.

Other Nonfunctional Requirements

- Performance Requirements: WebDojo DPS ensures smooth real-time code updates and preview rendering, optimizing performance for a seamless user experience.
- Safety Requirements: The application implements security best practices to protect user data and prevent unauthorized access.
- Security Requirements: WebDojo DPS adheres to security standards and guidelines, mitigating potential risks such as cross-site scripting (XSS) and injection attacks.
- Software Quality Attributes: The application prioritizes user-friendliness, responsiveness, and reliability to enhance user satisfaction.

Screen Shots-



```
<h1>Pranjal Kumar Dwivedi</h1>
```

```
h1{text-align:center;background-color:black;color:white;}
```

```
console.log("hello DPS");
```

```
Pranjal Kumar Dwivedi
```

Advantages of Project

WebDojo DPS offers several advantages, including:

- Simplified web development workflow
- Real-time code preview for immediate feedback
- Enhanced productivity and efficiency for developers

Chapter 5: Conclusion & Future Scope

Conclusion

In conclusion, WebDojo DPS represents a significant advancement in web development tools, providing a cohesive environment for code editing and previewing. Its user-friendly interface and real-time feedback capabilities make it a valuable asset for developers.

Future Scope

Future enhancements for WebDojo DPS may include:

- Integration of version control systems for collaborative development.
- Addition of advanced debugging and profiling tools.
- Expansion of supported languages and frameworks to cater to diverse developer needs.

References

1. MDN Web Docs: Mozilla's comprehensive resource for web developers, offering documentation, tutorials, and guides on HTML, CSS, and JavaScript.

    - Website: MDN Web Docs

2. W3Schools: A popular online learning platform providing tutorials, references, and examples for web development technologies including HTML, CSS, JavaScript, and more.

    - Website: W3Schools

3. GSAP (GreenSock Animation Platform) Documentation: Official documentation for GSAP, a powerful JavaScript animation library used for creating smooth animations in WebDojo DPS.

    - Website: GSAP Documentation

4. Agile Methodology: A guide to Agile software development principles and practices, essential for understanding the iterative and adaptive approach adopted in the development of WebDojo DPS.

- Book: "Agile Software Development: Principles, Patterns, and Practices" by Robert C. Martin

5. User Interface Design Principles: A reference for principles of user interface design, ensuring that WebDojo DPS provides an intuitive and user-friendly experience.

    - Book: "Don't Make Me Think: A Common Sense Approach to Web Usability" by Steve Krug

6. Version Control with Git: A guide to version control using Git, which could be integrated into WebDojo DPS for collaborative development and version management.

    - Book: "Pro Git" by Scott Chacon and Ben Straub

7. Web Development Best Practices: A collection of best practices and guidelines for web development, ensuring that WebDojo DPS adheres to industry standards.

    - Website: Google Web Fundamentals

These references cover a wide range of topics relevant to the development of WebDojo DPS, including programming languages, development methodologies, animation libraries, user interface design, version control, and web development best practices.

User Manual

Welcome to WebDojo DPS, your integrated web development environment for writing, editing, and previewing HTML, CSS, and JavaScript code in real-time. This user manual will guide you through the installation, setup, and usage of WebDojo DPS, ensuring a smooth and productive coding experience.

Table of Contents

1. Installation <a name="installation"></a>

To use WebDojo DPS, simply open your preferred web browser and navigate to the WebDojo DPS website. There is no need for installation or downloads, as WebDojo DPS runs entirely within your browser.

2. Getting Started <a name="getting-started"></a>

Upon loading the WebDojo DPS website, you will be presented with a user interface divided into three sections:

- HTML editor
- CSS editor
- JavaScript editor

You can start writing and editing your code directly in these editors.

3. Writing Code <a name="writing-code"></a>

Use the HTML, CSS, and JavaScript editors to write your code. The editors provide syntax highlighting and code formatting to help you write clean and organized code.

4. Previewing Output <a name="previewing-output"></a>

As you write and edit your code, the output preview window will automatically update to show you the rendered output of your HTML, CSS, and JavaScript code in real-time. This allows you to see how your changes affect the appearance and behavior of your web page instantly.

5. Saving Code <a name="saving-code"></a>

WebDojo DPS does not require you to create an account or save your code manually. However, you can copy your code from the editors and paste it into a text editor of your choice to save it locally on your computer.

6. Keyboard Shortcuts <a name="keyboard-shortcuts"></a>

WebDojo DPS supports the following keyboard shortcuts for convenient code editing:

- **Ctrl + S**: Save code
- **Ctrl + Z**: Undo
- **Ctrl + Y**: Redo

7. Troubleshooting <a name="troubleshooting"></a>

1. Refresh the page.
2. Clear your browser's cache and cookies.
3. Try using a different web browser.