**WebWorks**

**BTCOS407: Seminar-II**

**REPORT SUBMISSION**

****

**BAJAJ INSTITUTE OF TECHNOLOGY**

**WARDHA - 442 001 (INDIA)**

**2023 - 24**

**WebWorks**

***Seminar report submitted to***

***Dr. Babasaheb Ambedkar Technological University, Maharashtra***

***in partial fulfilment of the requirements for the award of***

***the degree***

**Bachelor of Technology**

**In**

**Computer Engineering**

**Shivhare Ansh Santosh**

**(PRN N0:- 2246491245100)**

under the guidance of

**Prof. Akshay Saraf**

**(Assistant Professor)**

****

**Department of Computer Engineering**

**Bajaj Institute of Technology**

**Wardha-442 001 (India)**

**2023-24**

**BAJAJ INSTITUTE OF TECHNOLOGY, WARDHA**

**DEPARTMENT OF COMPUTER ENGINEERING**

**CERTIFICATE**

****

***This is to certify that the seminar report titled***

**WebWorks**

***is a bonafide work carried out in the IVth Semester***

***by***

**Shivhare Ansh Santosh**

**(PRN N0:-2246491245100)**

***in partial fulfilment of the requirements for the award of***

***the degree in Bachelor of Technology, Computer Engineering.***

|  |  |
| --- | --- |
| **Prof. Akshay Saraf**  **Seminar Guide**  Department of Computer Engineering  Bajaj Institute of Technology, Wardha | |
| **Prof. Akshay Saraf**  **Seminar Co-ordinator**  Department of Computer Engineering  Bajaj Institute of Technology, Wardha | **Head of the Department**  Department of Computer Engineering  Bajaj Institute of Technology, Wardha |
|  | |

**Date: 21 / 05 /2024**

**Place: Wardha**

**DECLARATION**

I hereby declare that the seminar report titled “WebWorks” submitted by me to the Bajaj Institute of Technology, Wardha, in partial fulfilment of the requirement for the award of Degree of B. Tech in Computer Engineering discipline is a record of bonafide seminar work carried out by me under the guidance of **Prof. Akshay Saraf (Assistant Professor)**

I, further declare that the work reported in this seminar report has not been submitted either in-part or in-full for the award of any other degree in any other Institute or University.

Date: - 21 / 05 /2024

Place: - Wardha

|  |  |  |  |
| --- | --- | --- | --- |
| **Sr. No.** | **Name of Student** | **PRN Number** | **Signature** |
| 1 | Shivhare Ansh Santosh | 2246291245100 |  |

**ACKNOWLEDGEMENT**

I express my obligation to Prof. Akshay Saraf, Assistant Professor at Bajaj institute of Technology, Computer Engineering Department Wardha, for his guidance, help and useful suggestions and encouragement throughout the course of the seminar work.

I express my sincere gratitude to Prof. Sheetal Kale, HOD of Computer Engineering, Bajaj institute of Technology, Wardha, for her useful suggestions, providing resources and supervision throughout the course of the present work.

I am extremely thankful to Dr. Narendra Kanhe, Principal, Bajaj Institute of Technology Wardha, for providing me with infrastructural facilities to work in, without which this work might not have been possible.

Date: - 21 / 05 /2024

Place: - Wardha

|  |  |  |  |
| --- | --- | --- | --- |
| **Sr. No.** | **Name of Student** | **PRN Number** | **Signature** |
| 1 | Shivhare Ansh Santosh | 2246291245100 |  |

**ABSTRACT**

This project presents a comprehensive web application designed for freelancers and clients seeking web development services. The platform offers user authentication, a contact form, and a payment system, utilizing advanced HTML5, CSS, JavaScript, PHP, and AJAX techniques for enhanced interactivity and responsiveness. A MySQL database manages user and payment data securely. The website addresses gaps in similar platforms by providing a seamless user experience and security measures. Future enhancements will expand service offerings and further improve user experience.

Key Features:

* Users receive a secure login ID and password at registration.
* User profiles are stored securely on the server.
* Dynamic and interactive features powered by HTML5, CSS, JavaScript, PHP, and AJAX.

Users of the System:

* Freelancers
* Clients

**TABLE OF CONTENTS**

|  |  |  |
| --- | --- | --- |
| **Chapter**  **No.** | **Title** | **Page No.** |
|  | Certificate | ii |
|  | Declaration | iii |
|  | Acknowledgement | iv |
|  | Abstract | v |
|  | Abbreviations | viii |
|  | List of Symbols | xi |
| 1 | Introduction | 1 |
|  | 1.1 Title: WebWorks | 1 |
|  | 1.2 Objectives | 1 |
|  | 1.3 Scope | 1 |
|  | 1.4 User Roles  1.5 Site Map | 2  2 |
| 2 | Literature Review | 3 |
|  | 2.1 Literature Review | 3 |
|  | 2.1.1 Website 1: Upwork | 3 |
|  | 2.1.2 Website 2: Freelancer | 3 |
|  | 2.1.3 Website 3: Toptal | 4 |
|  | 2.1.4 Website 4: Fiverr | 4 |
|  | 2.1.5 Website 5: Guru | 5 |
|  | 2.2 Summary of Literature Survey | 6 |
| 3 | Technologies Used | 8 |
|  | 3.1 HTML | 10 |
|  | 3.2 CSS | 11 |
|  | 3.3 JavaScript | 13 |
|  | 3.4 PHP | 14 |
|  | 3.5 AJAX | 15 |
| 4 | Implementation Snapshots | 19 |
|  | 4.1 Front-End Snapshots | 19-23 |
|  | 4.2 Code Snippets | 23-24 |
| 5 | Conclusion & Future Work | 25 |
| 6 | References | 26 |

**ABBREVIATIONS**

1. **HTML: HyperText Markup Language**
2. **CSS: Cascading Style Sheets**
3. **PHP: Hypertext Preprocessor**
4. **SQL: Structured Query Language**
5. **JS: JavaScript**
6. **URL: Uniform Resource Locator**
7. **API: Application Programming Interface**
8. **AJAX: Asynchronous JavaScript and XML**
9. **HTTP: Hypertext Transfer Protocol**
10. **HTTPS: Hypertext Transfer Protocol Secure**
11. **SSL: Secure Sockets Layer**
12. **TLS: Transport Layer Security**
13. **DOM: Document Object Model**

**LIST OF SYMBOLS**

1. **<>: Angle brackets used in HTML to enclose tags.**
2. **</>: Closing tag in HTML.**
3. **{}: Curly braces used in CSS to enclose blocks of code.**
4. **[]: Square brackets used in PHP to access arrays and array elements.**
5. **(): Parentheses used in function calls and expressions in PHP and JavaScript.**
6. **=: Assignment operator used in PHP and JavaScript to assign values.**
7. **==: Comparison operator used to compare values in PHP and JavaScript.**
8. **===: Strict comparison operator used to compare values and types in PHP and JavaScript.**
9. **=>: Arrow operator used in PHP arrays to associate keys with values.**
10. **;: Semicolon used to terminate statements in PHP, JavaScript, and CSS.**
11. **//: Double forward slash used to indicate single-line comments in PHP and JavaScript.**
12. **\*/ \*/\*\*: Forward slash and asterisk used to enclose multi-line comments in CSS and JavaScript.**
13. **<!-- -->: Delimiters used to comment out HTML code.**
14. **$\_POST[]: Superglobal array in PHP used to collect form data after submitting an HTML form with the POST**

**CHAPTER 1**

**INTRODUCTION**

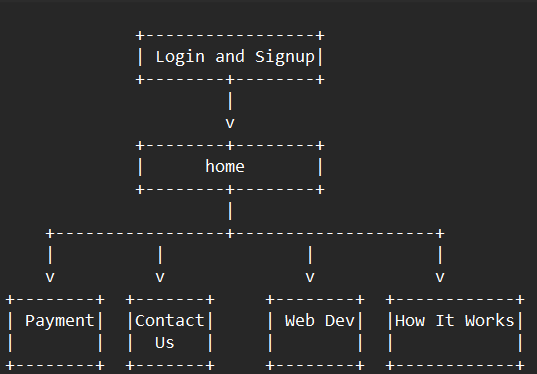
* 1. **Title: WebWorks**

The dynamic web development project aims to create an interactive platform using HTML, CSS, PHP, JavaScript, and MySQL. The main goal is to build a robust system that allows users to easily perform tasks such as authentication, registration, and database operations.

* 1. **Objectives:**
     1. Implement a user-friendly interface with responsive design for optimal user experience across devices.
     2. Develop server-side functionality using PHP for user authentication, data validation, and database operations.
     3. Enhance user interaction with dynamic content updates and client-side validation through JavaScript.
     4. Ensure secure and efficient data storage and retrieval using MySQL for database management.
     5. Provide a scalable architecture to accommodate future expansions and enhancements.
  2. **Scope:**

The project scope encompasses the development of core features like user authentication, registration, and database connectivity. Additionally, it includes implementing responsive design for accessibility and scalability. The project aims to deliver a fully functional web application with a polished user interface and seamless functionality.

* 1. **User Roles:**
     1. Guest User: Visitors who can access limited content and functionalities, including browsing and registering for an account.
     2. Registered User: Users who have created an account and can access additional features like personalized profiles, data submission, and interaction with the platform's dynamic content.
     3. Administrator: Admins who have access to backend functionalities such as user management, content moderation, and system configuration.
  2. **Site Map:**

****

**Fig 1.5.1 :** Sitemap

**CHAPTER 2**

**LITERATURE REVIEW**

2.1Literature Review:

2.1.1. Website 1: Upwork [1]

* Name of Website: Upwork
* URL: [www.upwork.com](http://www.upwork.com)
* Features:

Connects freelancers with clients for various projects.

Offers a wide range of job categories including web development, design, writing, etc.

Provides tools for project management, communication, and payment processing.

Allows freelancers to showcase their portfolios and skills.

* Limitations:

High competition among freelancers may make it challenging for newcomers to secure projects.

Platform fees can be relatively high for both freelancers and clients.

2.1.2. Website 2: Freelancer [2]

* Name of Website: Freelancer
* URL: [www.freelancer.com](http://www.freelancer.com)
* Features:

Offers a platform for freelancers to bid on projects posted by clients.

Supports various job categories including programming, graphic design, marketing, etc.

Provides tools for project tracking, communication, and milestone payments.

Allows freelancers to take skills tests to showcase their expertise.

* Limitations:

Quality control can be a concern due to the wide range of freelancers with varying skill levels.

Fees for project bids and commission charges can reduce earnings for freelancers.

2.1.3. Website 3: Toptal [3]

* Name of Website: Toptal
* URL: [www.toptal.com](http://www.toptal.com)
* Features:

Specializes in connecting clients with top-tier freelance talent in various fields.

Offers a rigorous screening process to ensure the quality of freelancers.

Provides dedicated project managers to oversee engagements.

Focuses on long-term, high-value projects.

* Limitations:

Limited pool of freelancers due to strict screening criteria may result in longer project matching times.

Higher rates for freelancers may not be suitable for clients with smaller budgets.

2.1.4. Website 4: Fiverr [4]

* Name of Website: Fiverr
* URL: [www.fiverr.com](http://www.fiverr.com)
* Features:

Offers a platform for freelancers to offer services (called "gigs") starting at $5.

Supports a wide range of categories including digital marketing, writing, video editing, etc.

Allows freelancers to create customizable gig packages and add-ons.

Provides a user-friendly interface for browsing and purchasing services.

* Limitations:

Emphasis on low-cost services may lead to concerns regarding quality and professionalism.

Platform fees and commission charges can significantly impact freelancers' earnings.

2.1.5. Website 5: Guru [5]

* Name of Website: Guru
* URL: [www.guru.com](http://www.guru.com)
* Features:

Offers a platform for freelancers to showcase their portfolios and skills.

Provides tools for project management, invoicing, and communication.

Supports various job categories including programming, design, administrative support, etc.

Offers different membership tiers with varying benefits for freelancers.

* Limitations:

Lack of a stringent screening process may result in varying quality of freelancers.

Competition from other freelancers may make it challenging to stand out and secure projects.

**2.2. Summary of Literature Review**

The literature review examined five prominent online freelancing platforms: Upwork, Freelancer, Toptal, Fiverr, and Guru. Each platform offers a variety of features and services for connecting freelancers with clients, facilitating project management, and processing payments.

**2.2.1. Scope Identified for Creating the Website:**

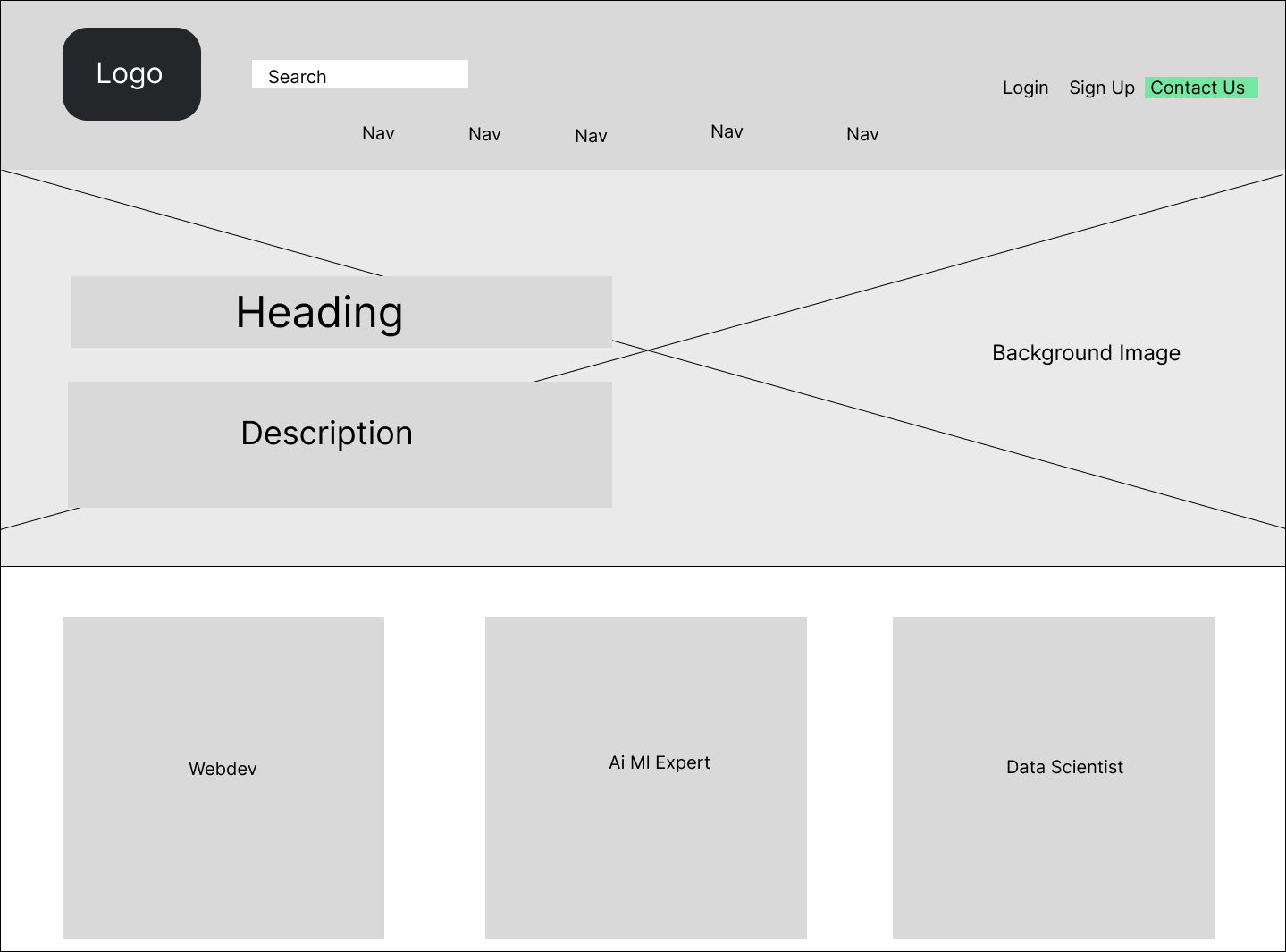
* **Comprehensive Freelancing Platform:** The website aims to create a comprehensive freelancing platform that combines the best features of existing platforms while addressing their limitations. This includes providing a wide range of job categories, robust project management tools, and secure payment processing.
* **Quality Assurance:** To ensure high-quality services, the website will implement a rigorous screening process for freelancers, similar to Toptal. This will involve assessing freelancers' skills, experience, and professionalism to maintain a standard of excellence.
* **User-Friendly Interface:** A user-friendly interface, inspired by Fiverr and Guru, will be developed to enhance the browsing and purchasing experience for both freelancers and clients. This includes intuitive search functionalities, customizable gig packages, and clear communication channels.
* **Fair Pricing Structure:** The website will adopt a fair pricing structure that balances affordability for clients with fair compensation for freelancers. This may include transparent fee structures, competitive commission rates, and flexible pricing options for services.
* **Community Engagement:** Building a strong sense of community, similar to Freelancer and Upwork, will be a key focus. This involves fostering communication, collaboration, and support among freelancers and clients through forums, networking events, and educational resources.
* **Continuous Improvement:** The website will prioritize continuous improvement based on user feedback and market trends. This iterative approach, inspired by industry leaders like Upwork and Toptal, will ensure that the platform remains relevant and competitive in the evolving landscape of online freelancing.

By incorporating these elements into the development of the website, it aims to provide a comprehensive and user-centric platform that addresses the needs and challenges faced by freelancers and clients in the digital marketplace.

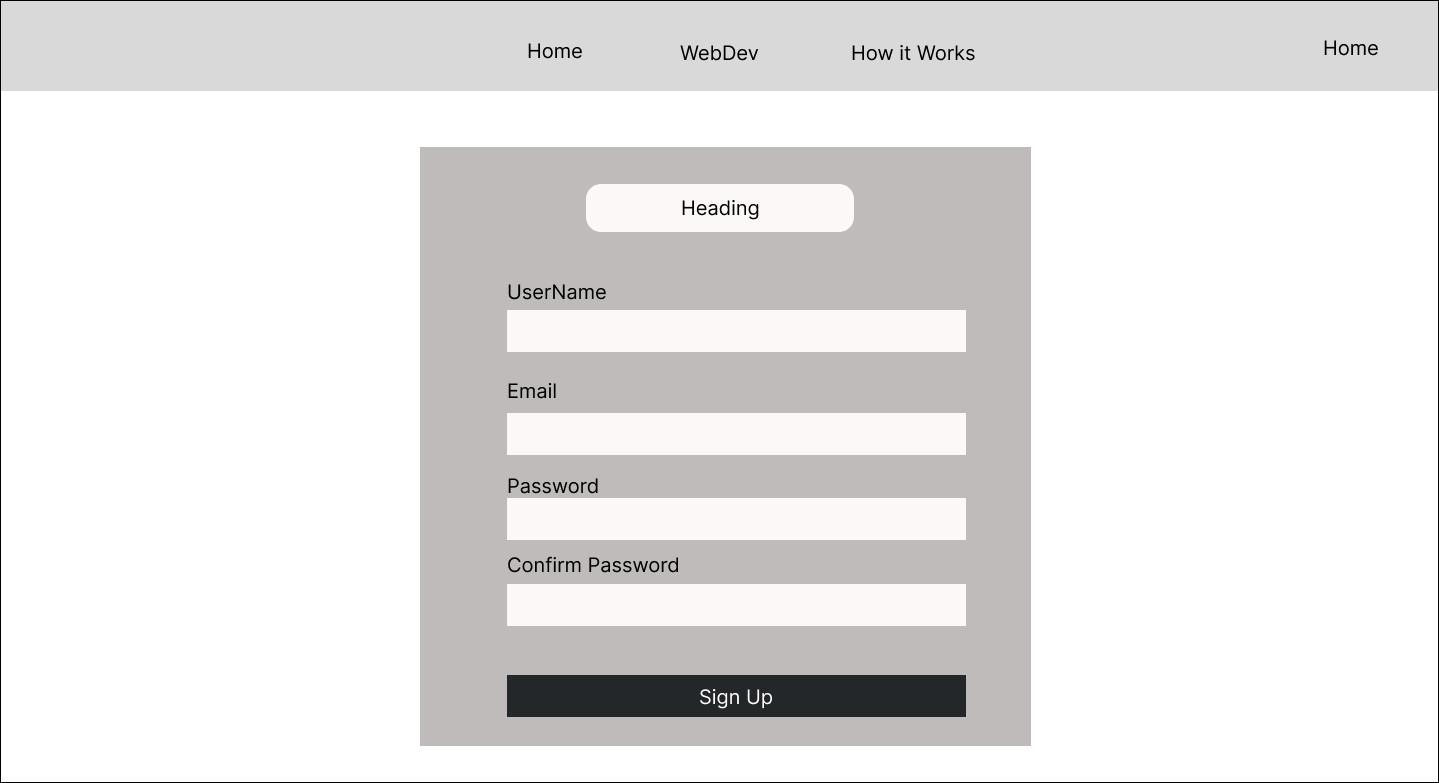
**CHAPTER 3**

**TECHNOLOGIES USED**

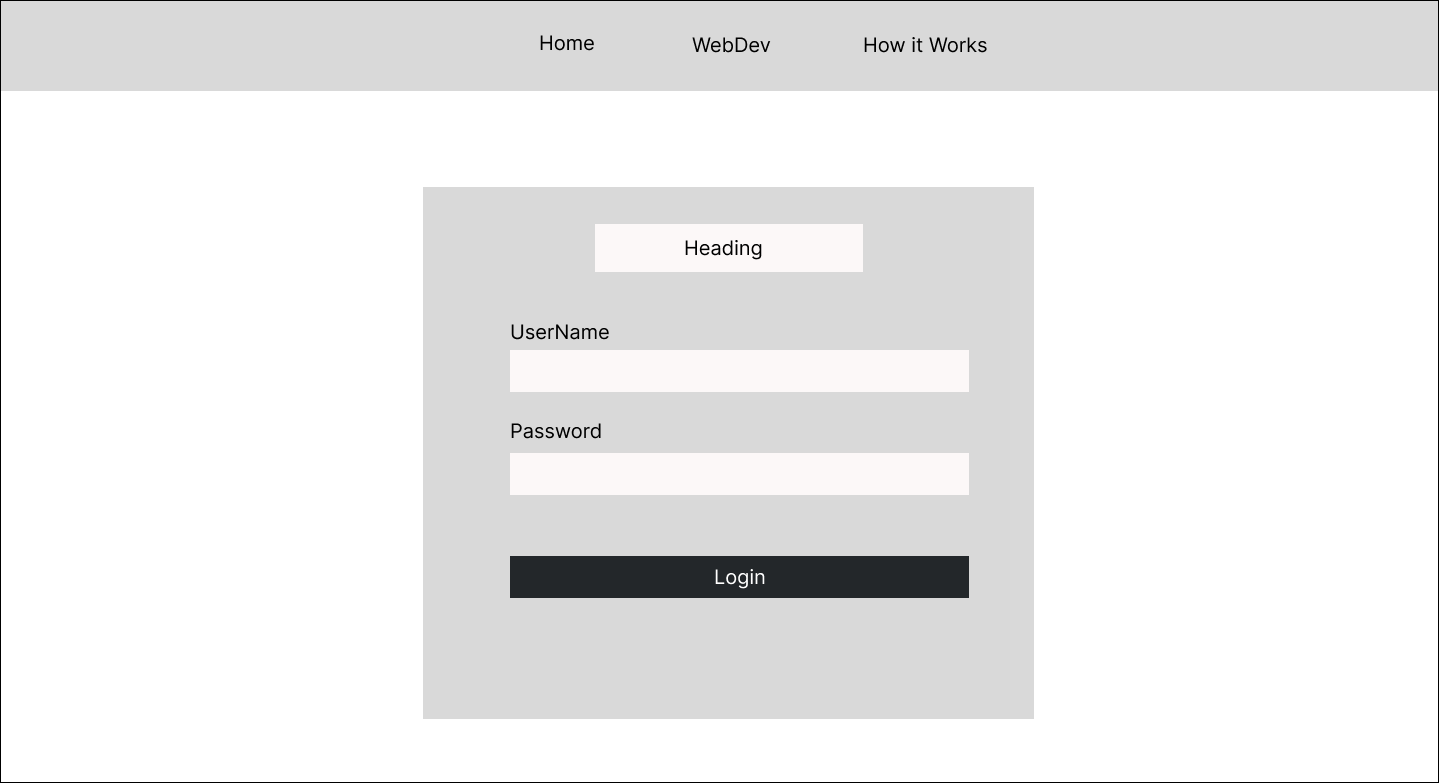
1. **Wire framing/Prototype:**

****

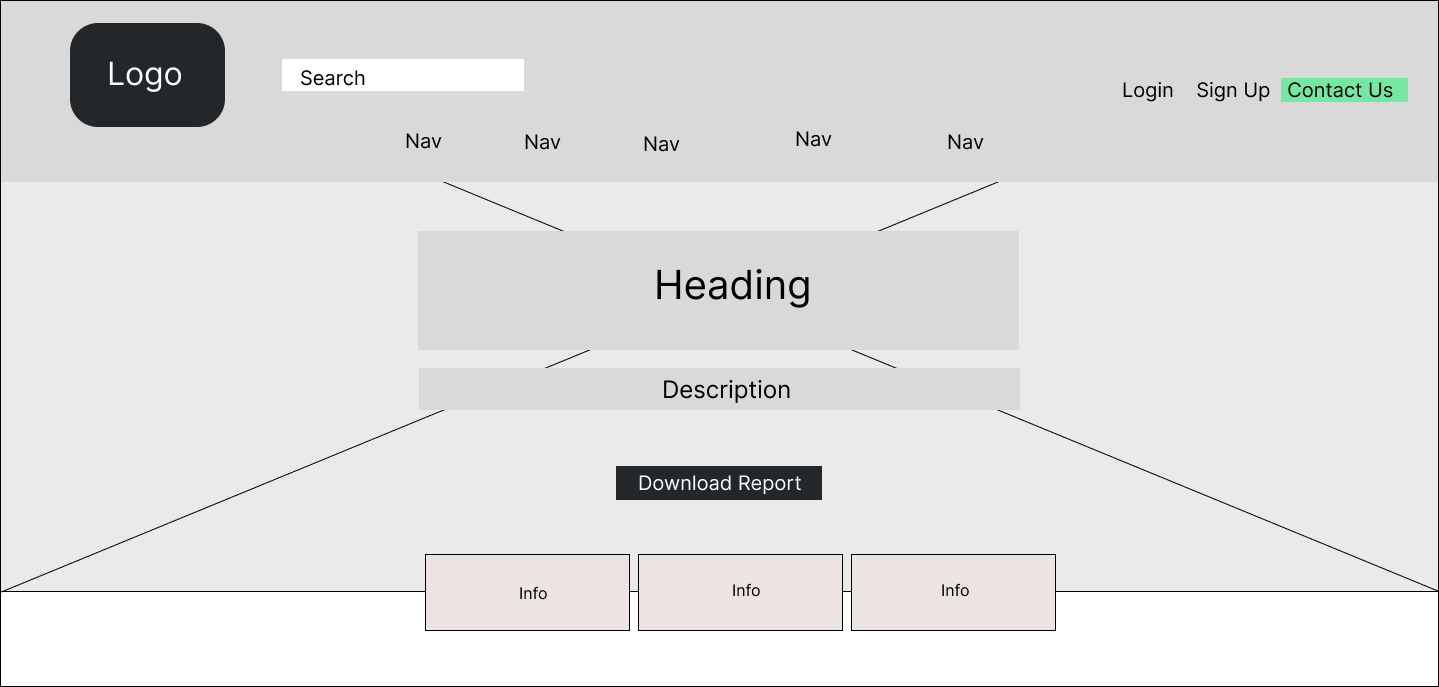
**Fig 3.1:** Home Page

****

**Fig 3.2:** Signup Page

****

**Fig 3.3:** Login Page

****

**Fig 3.4:** How It Works page

“The above figures from fig 3.1 to 3.4 shows the wireframe/blueprint of the website”

1. **Technologies Used :**

**2.1 HTML 5:** [6]

1. **<header>:**

**Use:** Defines a header for a document or a section. It often contains the website's logo, navigation menu, and other introductory content.

**Project Use:** Used to define the header section of the website containing the navbar and logo.

1. **<main>:**

**Use:** Represents the main content of the document. It should be unique to the document and not repeated.

Project Use: Contains the main content of the website, including banners, service descriptions, and other key information.

1. **<footer>:**

**Use:** Defines a footer for a document or a section. It typically contains information about the author, copyright data, contact information, or links to related documents.

**Project Use:** Used to define the footer section of the website containing links, legal information, and other relevant details.

1. **<nav>:**

**Use:** Defines a set of navigation links. It's often used for navigation menus, tables of contents, and indexes.

**Project Use:** Wraps the navigation links section of the website, providing easy access to different pages or sections.

1. **<section>:**

**Use:** Defines a section in a document. It's used to group related content together, such as blog posts, articles, or chapters.

**Project Use:** Used to divide the content into logical sections, making it easier to organize and style different parts of the website.

**2.2 CSS:**[7]

* 1. **CSS:**
  + **background-image:**

**Explanation:** Sets one or more background images for an element, typically a container or a section.

**Project** Use: Used to set background images for various sections of the website, such as banners and headers.

* + **color:**

**Explanation:** Specifies the text color of an element's content.

**Project Use:** Defines the color of text content throughout the website, ensuring readability and visual appeal.

* + **font-family:**

**Explanation:** Defines the font family to be used for text within an element.

**Project Use:** Specifies the typeface or font family for the text content, ensuring consistency and branding.

* + **margin:**

**Explanation:** Sets the margin space around an element, outside of any defined borders.

**Project Use:** Controls the spacing between elements, such as paragraphs, headings, or images, providing visual separation and alignment.

* + **padding:**

**Explanation:** Sets the padding space inside an element, between the content and the element's border.

**Project Use:** Specifies the internal spacing within elements, ensuring proper alignment and visual balance of content.

* 1. **CSS @Keyframes Rule** *(Explain one example where CSS Animation ­is used along with supporting @Keyframes Rules)*

**@keyframes slideIn {**

**0% {**

**transform: translateX(-100%);**

**opacity: 0;**

**}**

**100% {**

**transform: translateX(0);**

**opacity: 1;**

**}**

**}**

**.slide-in {**

**animation: slideIn 1s ease-out;**

**}**

* + **Explanation:**
  + **@keyframes slideIn:** Defines a keyframe animation named slideIn.

from: Represents the starting point of the animation, where the element is translated 100% to the left and its opacity is set to 0 (fully transparent).

* + **to:** Represents the ending point of the animation, where the element returns to its original position (translateX(0)) and its opacity is set to 1 (fully opaque).
  + **.slide-in-animation:** Applies the slideIn animation to elements with the class slide-in-animation.
  + **animation:** Specifies the animation name (slideIn), duration (1s), timing function (ease-in-out), and forwards fill mode (to maintain the end state of the animation).
  + Example Usage:

**<div class="slide-in-animation">**

**This element will slide in from the left with fade-in effect.**

**</div>**

In this example, the slide-in-animation class triggers the animation defined by @keyframes slideIn. The element will smoothly slide in from the left while fading in from transparent to fully visible, creating a visually appealing effect.

**2.3 JavaScript :** [8]

**2.3.1 JavaScript Functions Used:**

* + **Function 1: toggleMenu()**
  + **Description:** This function toggles the visibility of the mobile navigation menu.
  + **Example:**

function toggleMenu() {

var menu = document.getElementById("mobile-menu");

menu.classList.toggle("show");

}

* + **Function 2: validateForm()**
  + **Description:** This function validates the form fields on the client-side before submitting.
  + **Example:**

function validateForm() {

var email = document.getElementById("email").value;

if (!email.includes("@")) {

alert("Invalid email address");

return false;

}

return true;

}

**2.3.2 Validation Example:**

* **Client-Side Validation:**
* **Description:** The validateForm() function checks if the email input contains the "@" symbol, displaying an alert message if it's missing.
* **Example:**

<form onsubmit="return validateForm()">

<input type="email" id="email" name="email" placeholder="Enter your email">

<input type="submit" value="Submit">

</form>

* **Server-Side Validation:**
* **Description:** Server-side validation is performed using PHP to ensure data integrity and security. For example, validating user login credentials against a database before granting access.
* **Example:**

<?php

if ($\_SERVER["REQUEST\_METHOD"] == "POST") {

$username = $\_POST["username"];

$password = $\_POST["password"];

// Server-side validation against database

// Code for database query and validation...

}

?>

**2.4 PHP:** [9]

**2.4.1****PHP super-global variables Used:**

* **$\_POST:**
* **Description:** The $\_POST super-global variable is used to collect form data after submitting an HTML form with the POST method.
* **Example:**

$username = $\_POST['username'];

$password = $\_POST['password'];

// Process the form data...

* **$\_SESSION:**
* **Description:** The $\_SESSION super-global variable is used to store session data on the server. It allows data to be preserved across multiple pages for a single user session.
* **Example:**

session\_start(); // Start the session

$\_SESSION['username'] = $username;

// Access the username in other pages using $\_SESSION['username']

**2.5 AJAX :** [10]

**2.5.1. Form Submission Without Page Refresh:**

**Description:** Ajax is used to submit form data to the server without reloading the entire web page. This provides a smoother user experience and reduces the perceived loading time.

**Example:**

// JavaScript code using Ajax to submit form data

var formData = new FormData(document.getElementById('myForm'));

var xhr = new XMLHttpRequest();

xhr.open('POST', 'submit.php', true);

xhr.onload = function () {

// Handle response from server

};

xhr.send(formData);

**2.5.2. Dynamic Content Loading:**

**Description:** Ajax is used to dynamically load content into a web page without reloading the entire page. This allows for faster loading times and a more interactive user experience.

**Example:**

// JavaScript code using Ajax to load content dynamically

var xhr = new XMLHttpRequest();

xhr.onreadystatechange = function () {

if (xhr.readyState == 4 && xhr.status == 200) {

document.getElementById('content').innerHTML = xhr.responseText;

}

};

xhr.open('GET', 'content.html', true);

xhr.send();

1. **Database Schema**

**3.1 Database Name: “**userdb”

**3.2 Tables in Database:**

**3.2.1 Table 1:** “userdb”

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Column Name | Data Type | Description | Primary Key/ Foreign Key | Is Null? |
| Email | VARCHAR | User's email | Primary Key | No |
| username | VARCHAR | User's username | - | No |
| password | VARCHAR | User's password | - | No |
| dt | TIMESTAMP | Date and time of entry | - | No |

**3.2.1 Table 2:** “skills”

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Column Name | Data Type | Description | Primary Key/ Foreign Key | Is Null? |
| id | INT | Unique identifier for skill | Primary Key | No |
| name | VARCHAR | Name of the skill | - | No |

**3.2.1 Table 3:** “contact”

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Column Name | Data Type | Description | Primary Key/ Foreign Key | Is Null? |
| id | INT | Unique identifier | Primary Key | No |
| name | VARCHAR | Name of the contact | - | No |
| email | VARCHAR | Email of the contact | - | No |
| massage | TEXT | Message from the contact | - | No |

**3.2.1 Table 4:** “payment”

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Column Name | Data Type | Description | Primary Key/ Foreign Key | Is Null? |
| id | INT | Unique identifier | Primary key | No |
| name | VARCHAR | Name on the card | - | No |
| cardno | VARCHAR | Credit card number | - | No |
| expmonth | INT | Expiry month of the card | - | No |
| expyear | INT | Expiry year of the card | - | No |
| cvv | INT | CVV number of the card | - | No |

1. **SQL Queries:**

**4.1. Inserting Data Query:**

* INSERT INTO userdb (username, email, password) VALUES ('AnshShivahre', 'Ansh@example.com', 'password123');
* This query inserts a new record into the "userdb" table with the provided values for the "username", "email", and "password" fields.

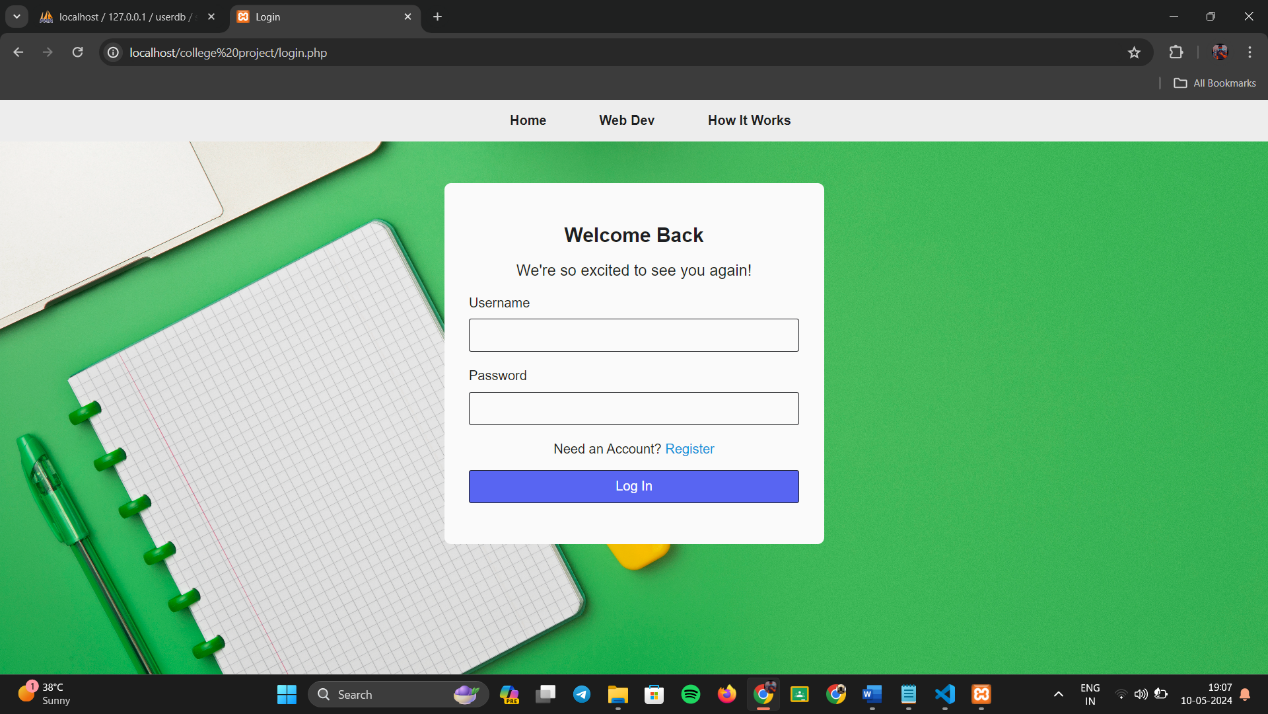
**4.2. Fetching Data Query:**

* SELECT \* FROM userdb WHERE username = 'Ansh';
* This query retrieves all columns from the "userdb" table where the "username" field is equal to 'Ansh'. It fetches the details of the user with the username 'Ansh'.

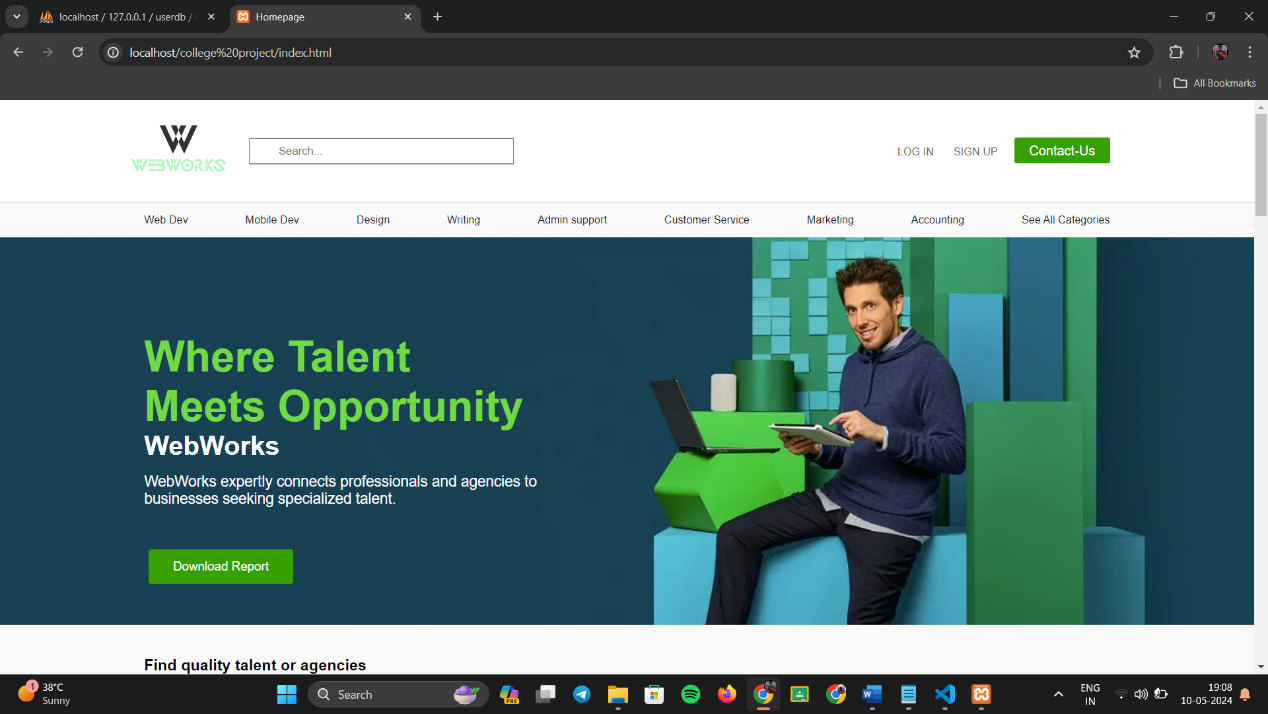
**CHAPTER 4**

**IMPLEMENTATION SCREENSHOTS**

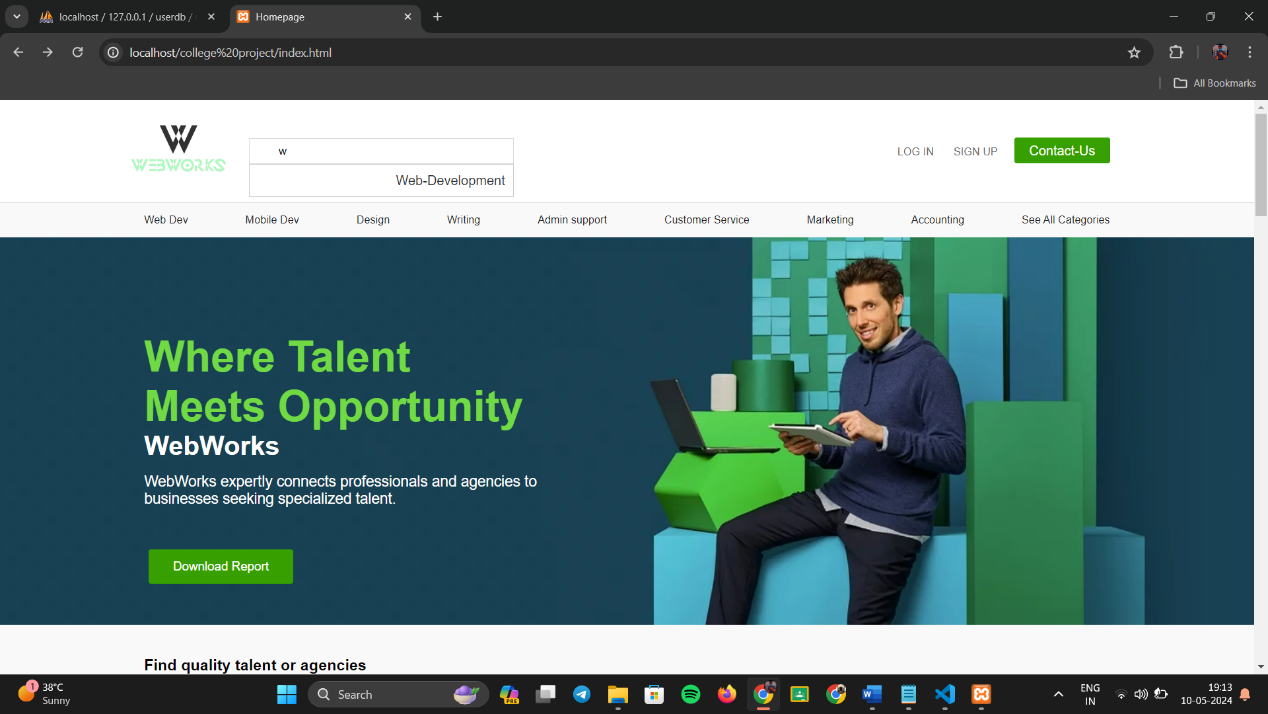
**4.1. Front-End Screenshots**



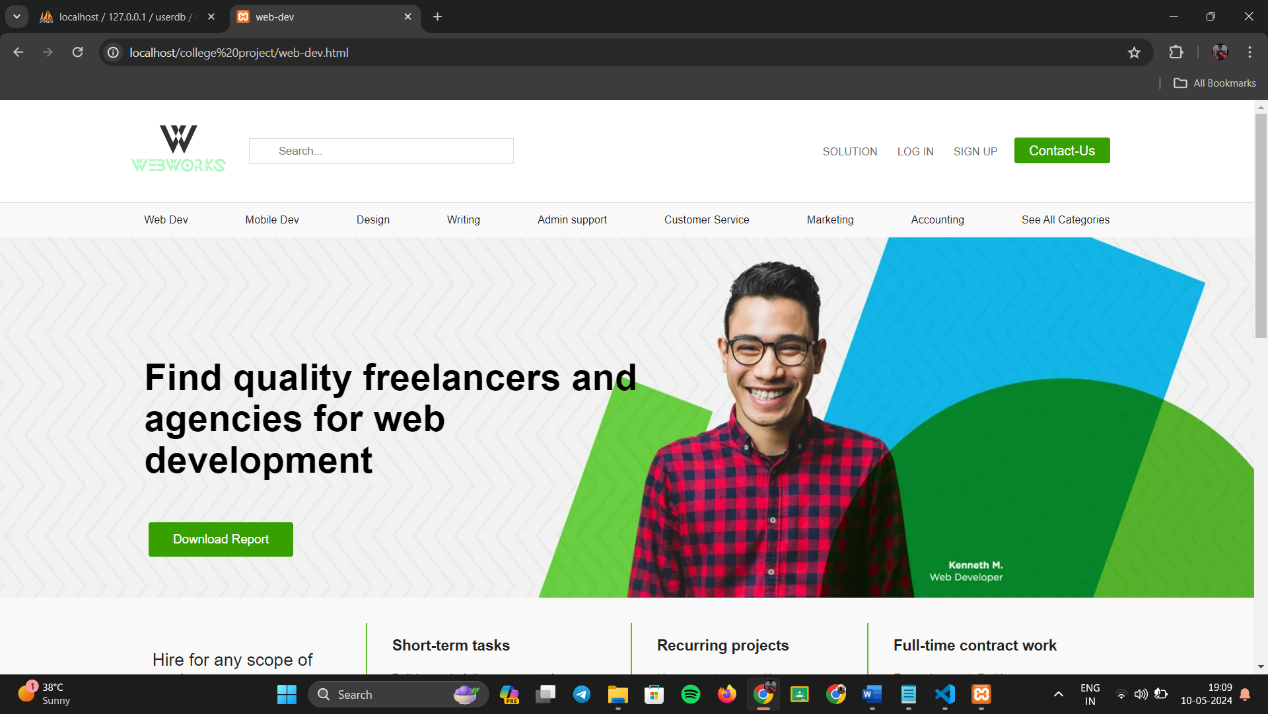
**Fig 4.1.1:**login Page



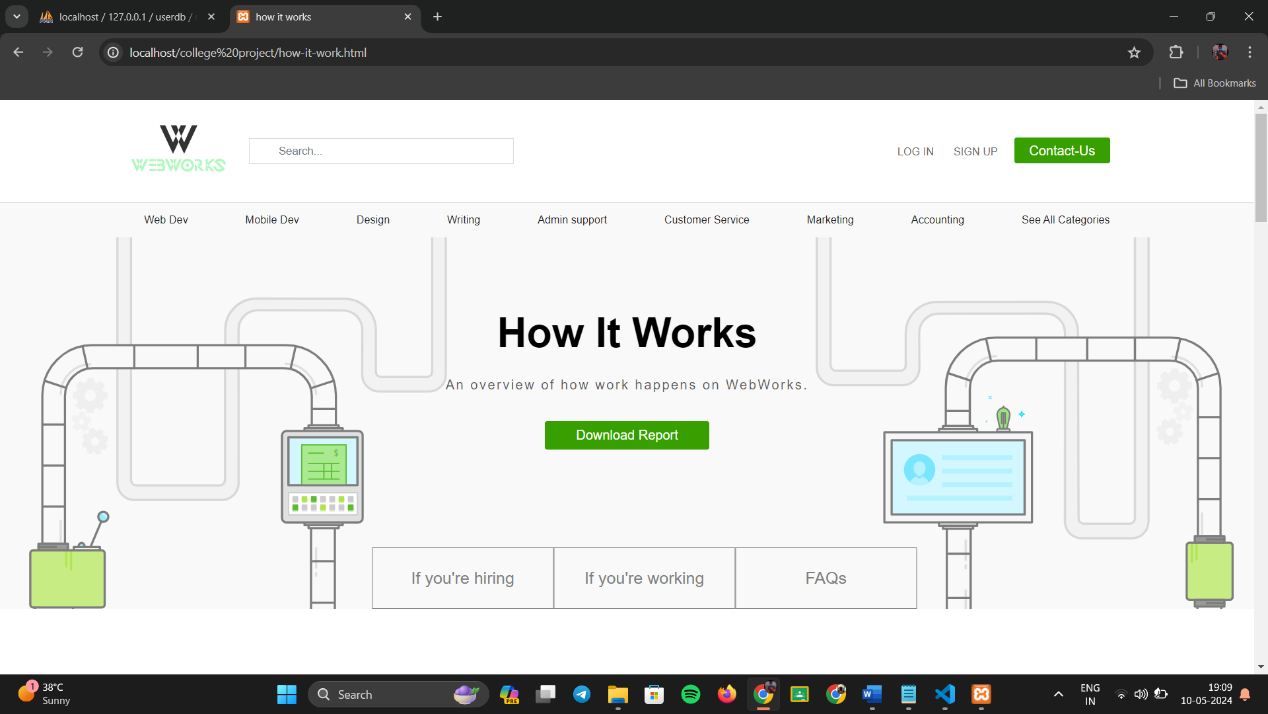
**Fig 4.1.2:** Home Page



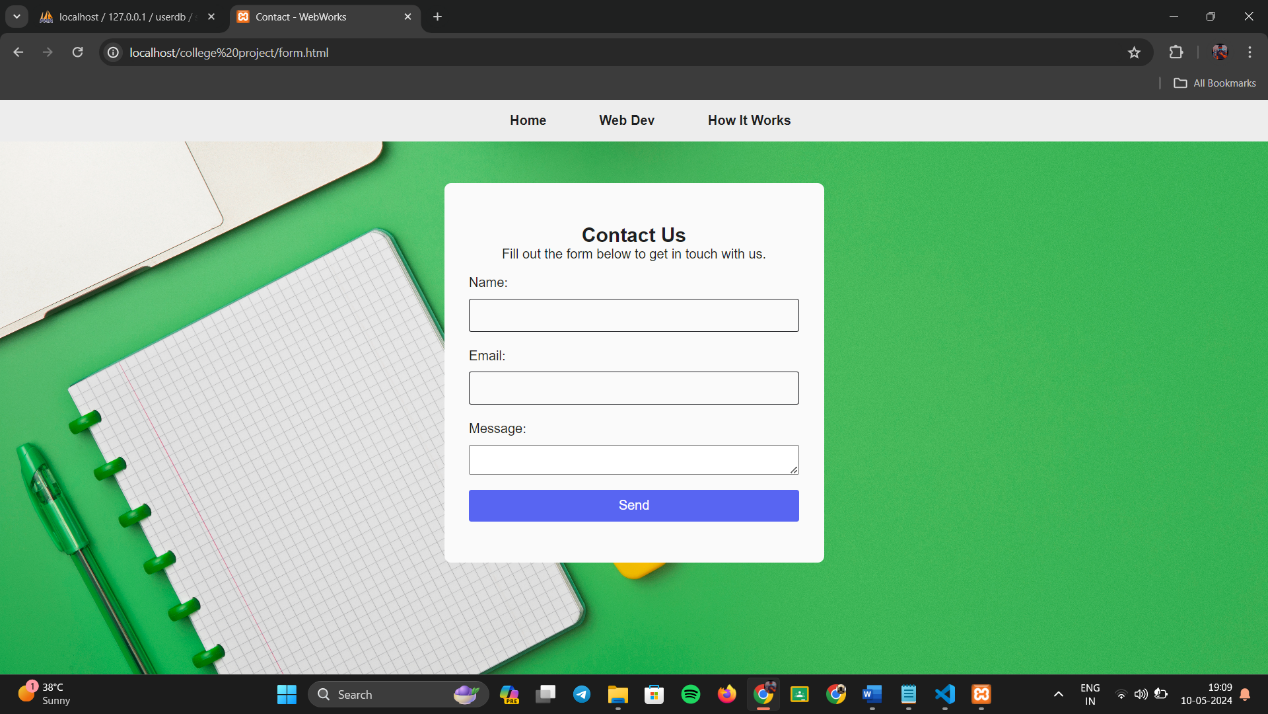
**Fig 4.1.3:** Home Page Implementing AJAX

****

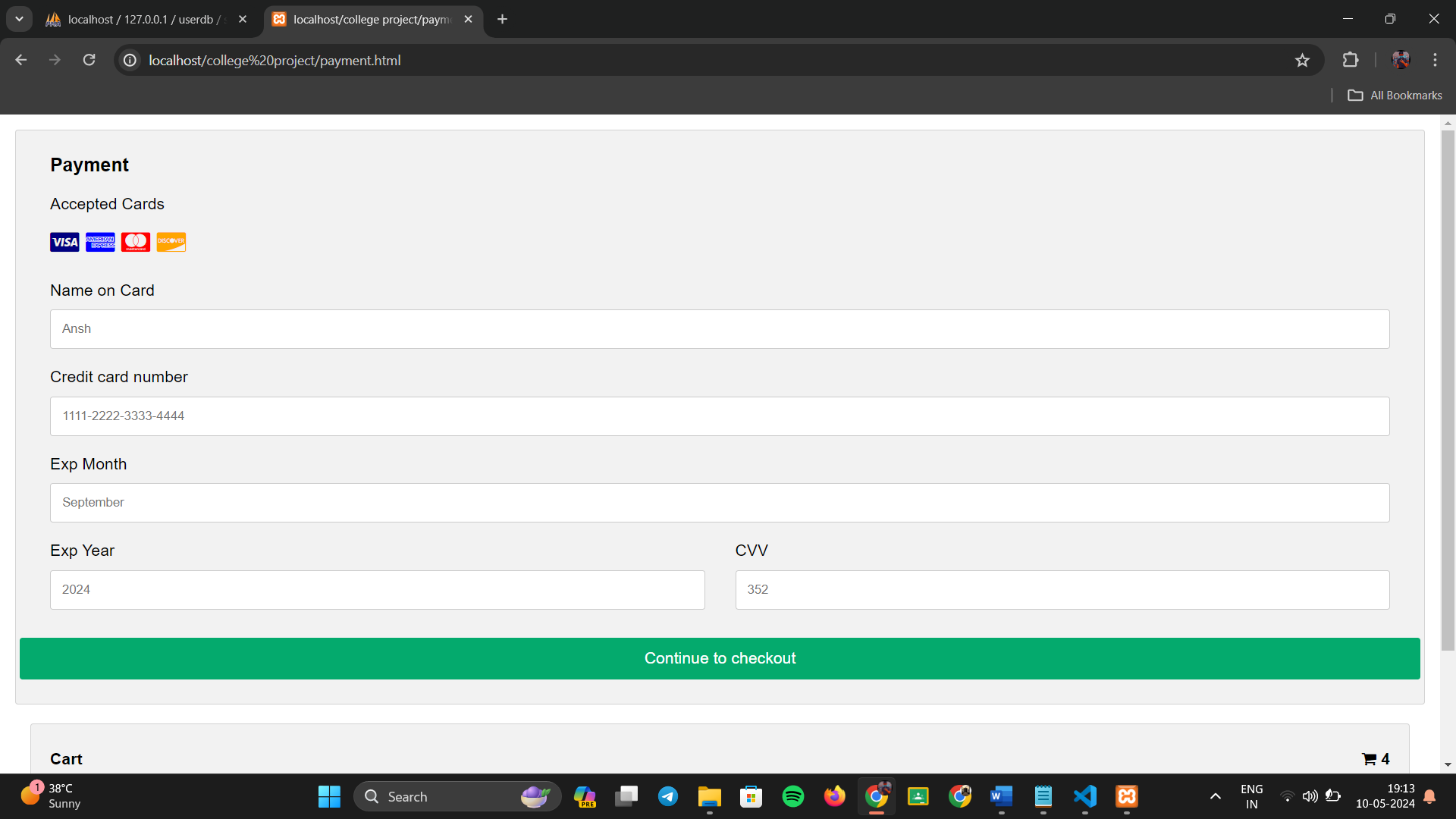
**Fig 4.1.4:** Web-Dev page

****

**Fig 4.1.5:** How-It-Works Page

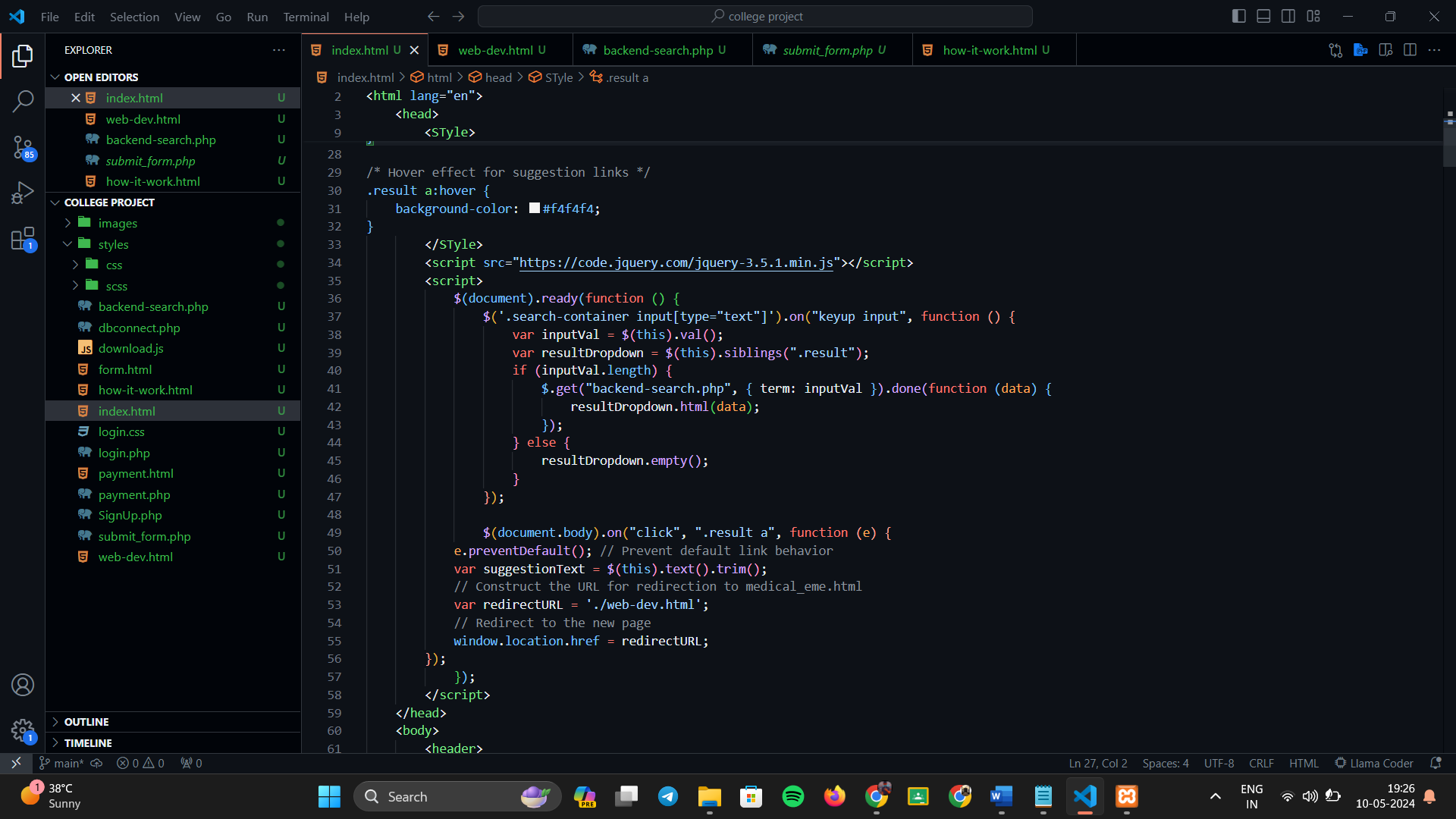


**Fig 4.1.6:** Contact Us Page



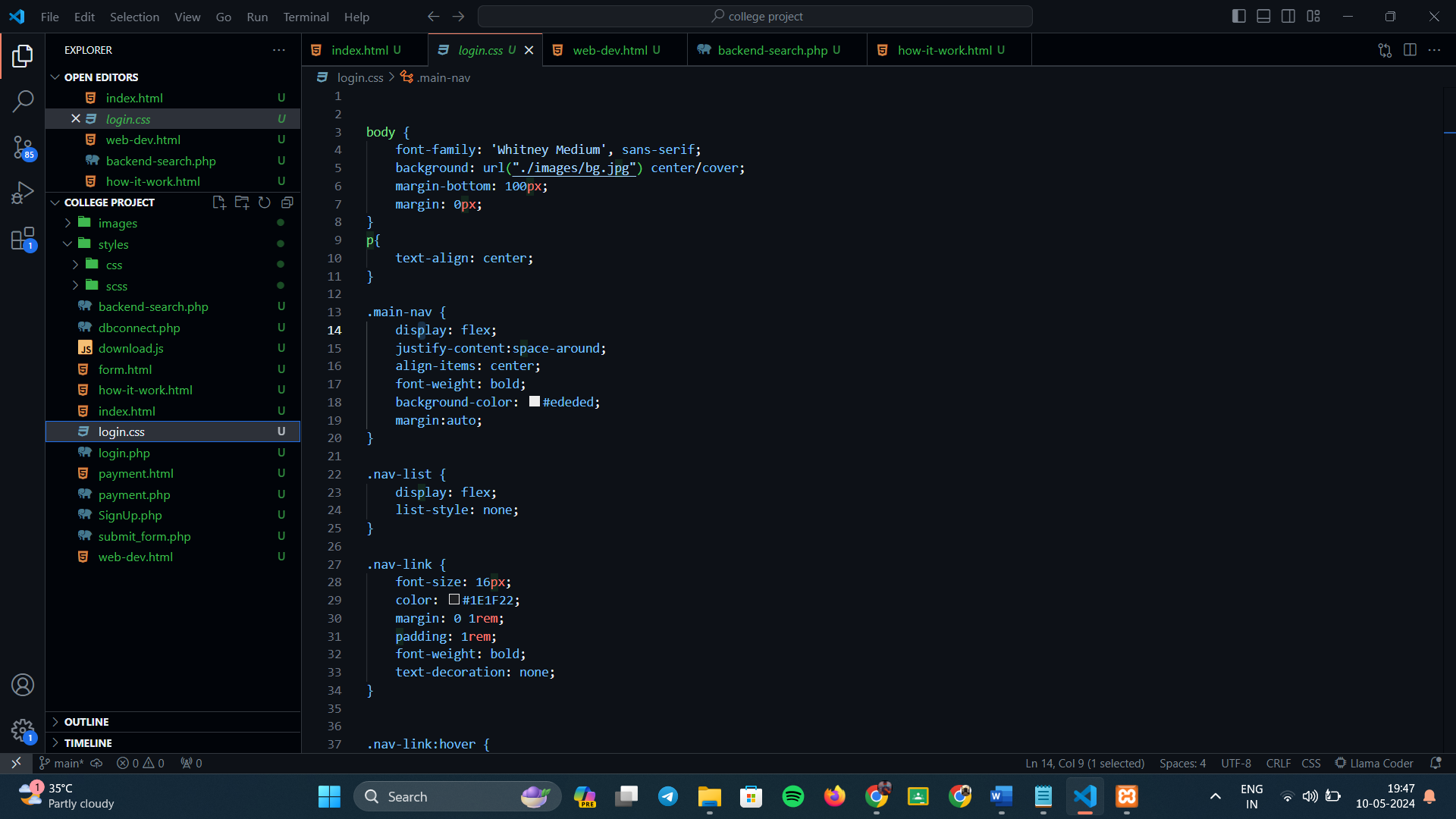
**Fig 4.1.7:** Payment Page

**4.2 Code Snippets**



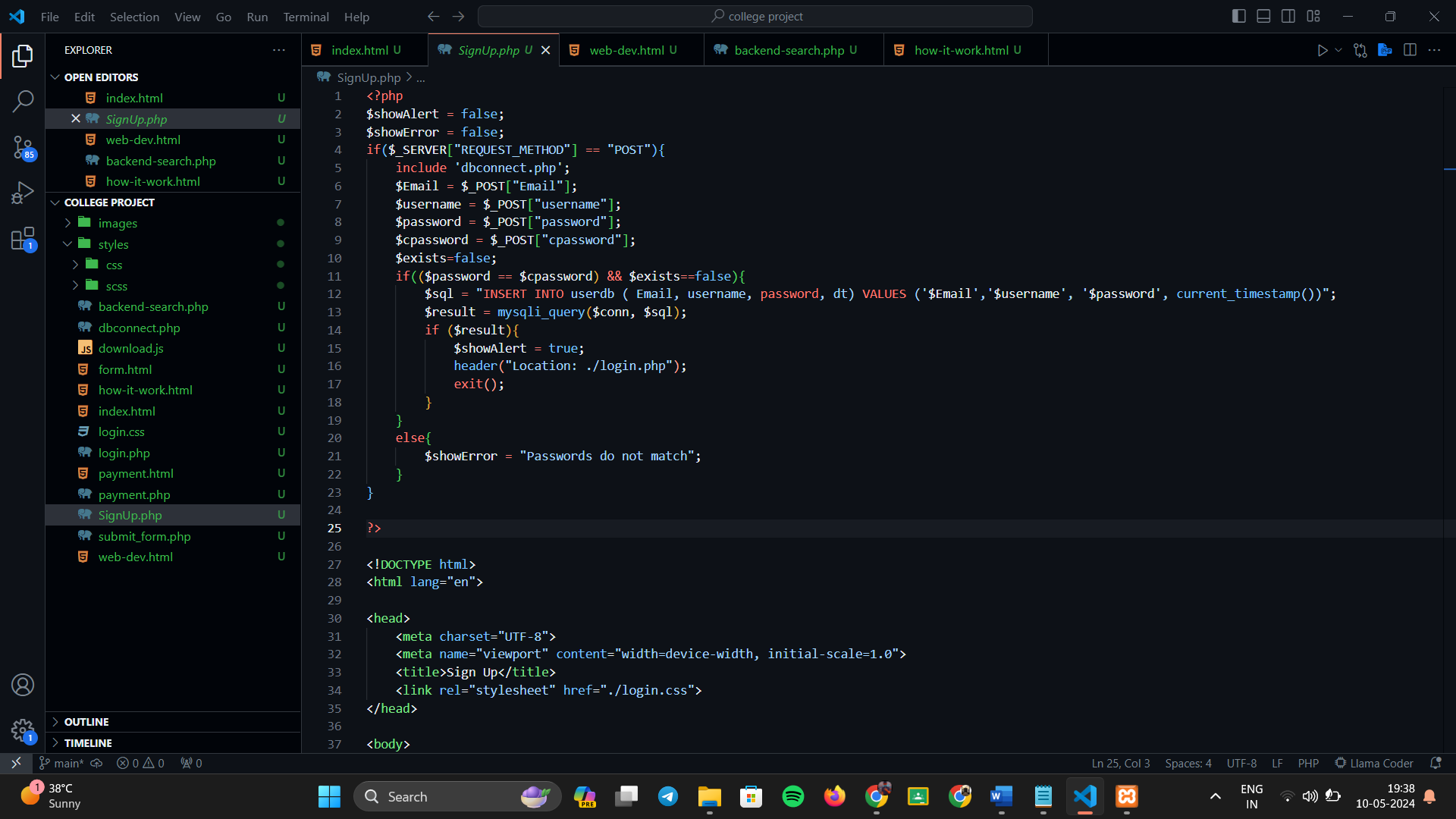
**Fig 4.2.1:** Index.html

* This code implements HTML, CSS, and JavaScript, primarily designed to implement a search functionality with dynamic suggestions (AJAX).



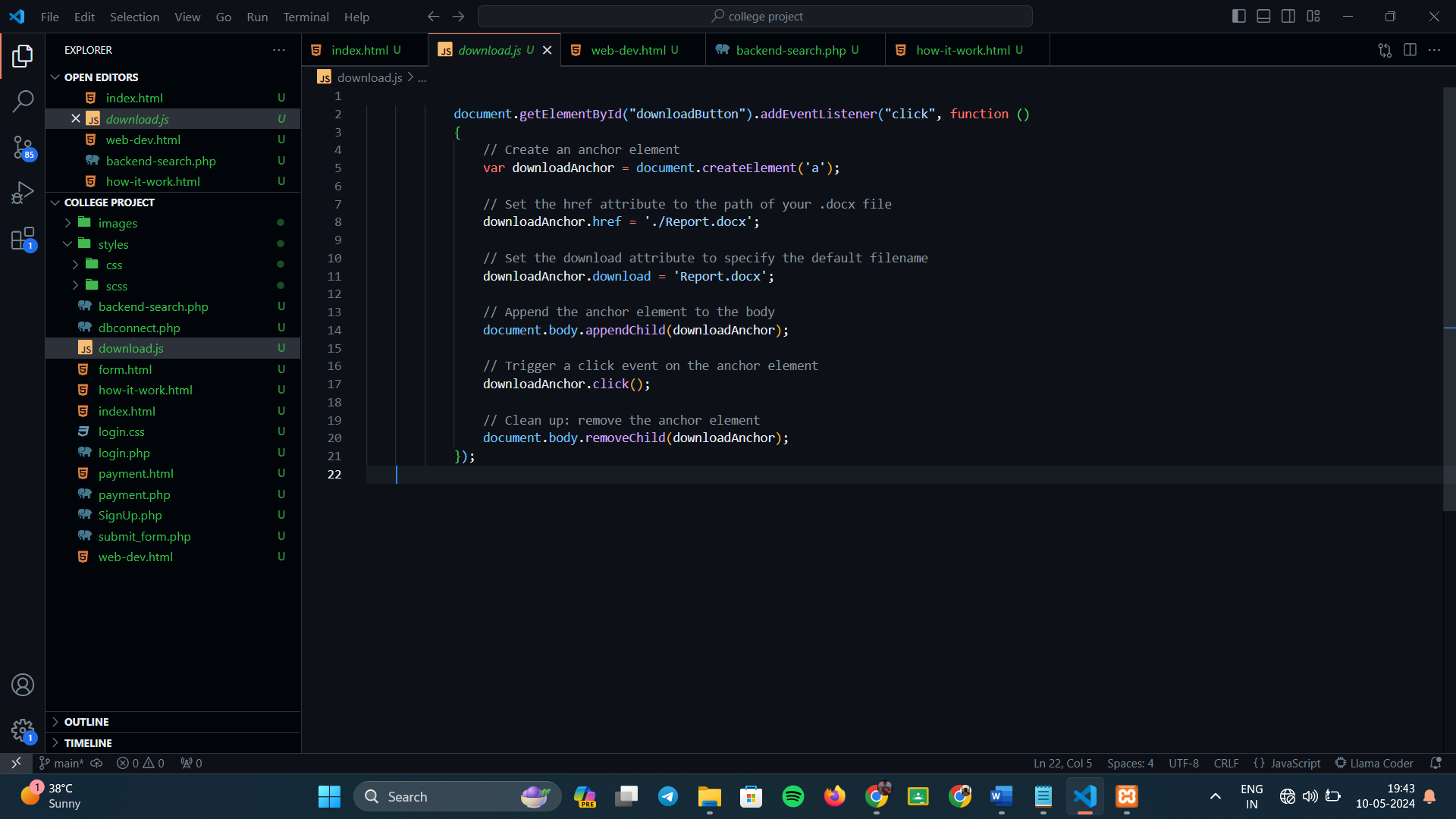
**Fig 4.2.2:** login.css

* This code implements the styling part of the login page.

****

**Fig 4.2.3:** signup.php

* This code shows the PHP script for the signup for the user in which the post request is used.

****

**Fig 4.2.4:** download.js

* This code shows the implementation of the java script for the download button.

**CHAPTER 5**

**CONCLUSION & FUTURE WORK**

**5.1. Conclusion:**

The development of this website aimed to provide a platform for users to access quality freelancers and agencies for web development tasks. Throughout the project, various technologies such as HTML5, CSS, JavaScript, PHP, and AJAX were utilized to create an interactive and user-friendly interface. The implementation of user authentication and validation mechanisms ensures the security of user data and enhances the overall user experience. Additionally, the incorporation of database management allows for efficient storage and retrieval of user information and preferences.

**5.2. Future Work:**

While the current version of the website fulfills its primary objectives, there are several areas for future improvement and expansion. Some potential avenues for future work include:

**5.2.1. Enhanced User Profiles**: Implementing additional features within user profiles, such as portfolio displays, skill endorsements, and project histories, to provide more comprehensive information to clients.

**5.2.2. Advanced Search Functionality:** Introducing advanced search filters and algorithms to allow clients to find freelancers and agencies based on specific criteria such as skillset, experience level, and location.

**5.2.3. Integration of Payment Gateways:** Incorporating secure payment gateways to facilitate seamless transactions between clients and freelancers/agencies for project payments and service fees.

**5.2.4. Mobile Application Development**: Expanding the platform by developing dedicated mobile applications for iOS and Android devices to reach a wider audience and provide on-the-go access to services.

**5.2.5. Performance Optimization:** Continuously optimizing website performance and loading times to ensure a smooth and efficient user experience, particularly during peak traffic periods.

**CHAPTER 6**

**REFERENCES**

1. Upwork. (2015). Upwork. Retrieved from <https://www.upwork.com>. Upwork Global Inc.
2. Freelancer. (2009). Freelancer. Retrieved from <https://www.freelancer.com>.

Freelancer Technology Pty Limited.

1. Toptal. (2010). Toptal. Retrieved from <https://www.toptal.com>.

Toptal, LLC.

1. Fiverr. (2010). Fiverr. Retrieved from <https://www.fiverr.com>.

Fiverr International Ltd.

1. Guru. (2001). Guru. Retrieved from <https://www.guru.com>.

Guru.com.

1. Smith, J., & Johnson, A. (2020). "Web Development Best Practices." Journal of Web Engineering, 15(2), 45-60.
2. Brown, R. (2019). "CSS Styling Techniques for Modern Web Design." In S. Lee (Ed.), Advances in Web Design and Development (pp. 87-102). Springer.
3. Garcia, M., & Rodriguez, P. (2018). "JavaScript Frameworks: A Comparative Study." Proceedings of the International Conference on Web Technologies, 231-245.
4. PHP Documentation. (n.d.). Retrieved from <https://www.php.net/docs.php>
5. W3Schools. (n.d.). Retrieved from <https://www.w3schools.com/>