**DEVELOPMENT AND DEPLOYMENT OF A PORTFOLIO WEBSITE USING DJANGO, HTML, CSS, JAVASCRIPT, AND MYSQL**

**Table of Contents**

[Introduction 4](#_Toc189485772)

[Project Overview 4](#_Toc189485773)

[Website Structure and Pages: 4](#_Toc189485774)

[Key Features of the Website: 8](#_Toc189485775)

[Backend Development with Django 9](#_Toc189485776)

[Django Setup: 9](#_Toc189485777)

[Database Integration (MySQL): 9](#_Toc189485778)

[Django Views and Templates: 10](#_Toc189485779)

[Frontend Development with HTML and CSS 11](#_Toc189485780)

[HTML Structure: 11](#_Toc189485781)

[CSS for Styling: 12](#_Toc189485782)

[3D and Dynamic Effects: 13](#_Toc189485783)

[JavaScript for Dynamic and Interactive Elements 14](#_Toc189485784)

[Form Validation: 14](#_Toc189485785)

[Interactive Features: 15](#_Toc189485786)

[Enhancing User Experience: 15](#_Toc189485787)

[Database Integration and MySQL 16](#_Toc189485788)

[Database Setup: 16](#_Toc189485789)

[Data Storage: 17](#_Toc189485790)

[Querying and Displaying Data: 17](#_Toc189485791)

[Handling User Submissions: 17](#_Toc189485792)

[Challenges and Solutions 17](#_Toc189485793)

[Testing and Deployment 19](#_Toc189485794)

[Conclusion 20](#_Toc189485795)

[References 21](#_Toc189485796)

Consolidate Tasks……………………………………………………………………………..27

# Introduction

This report provides an outline of the development procedure for a personal portfolio website designed to exhibit expert abilities, painting experience, and finished projects. The website serves as a dynamic platform to display the person's technical and tender talents, career achievements, and innovative projects in a visually attractive and interactive way. The goal of the portfolio website is to create an interesting online presence that highlights the individual’s qualifications, even providing visitors with smooth access to contact and challenge information.

The internet site has evolved the use of a mixture of contemporary net technology, which includes Python and Django for backend development, allowing efficient data management and user interactions. The **HTML** and **CSS** are employed for structuring and styling the frontend, ensuring a smooth and responsive design. **JavaScript** is included for dynamic functions together with form validation and interactive factors, whilst **MySQL** is applied for database management to shop person inputs, contact shape submissions, and portfolio undertaking facts.

# Project Overview

## Website Structure and Pages:

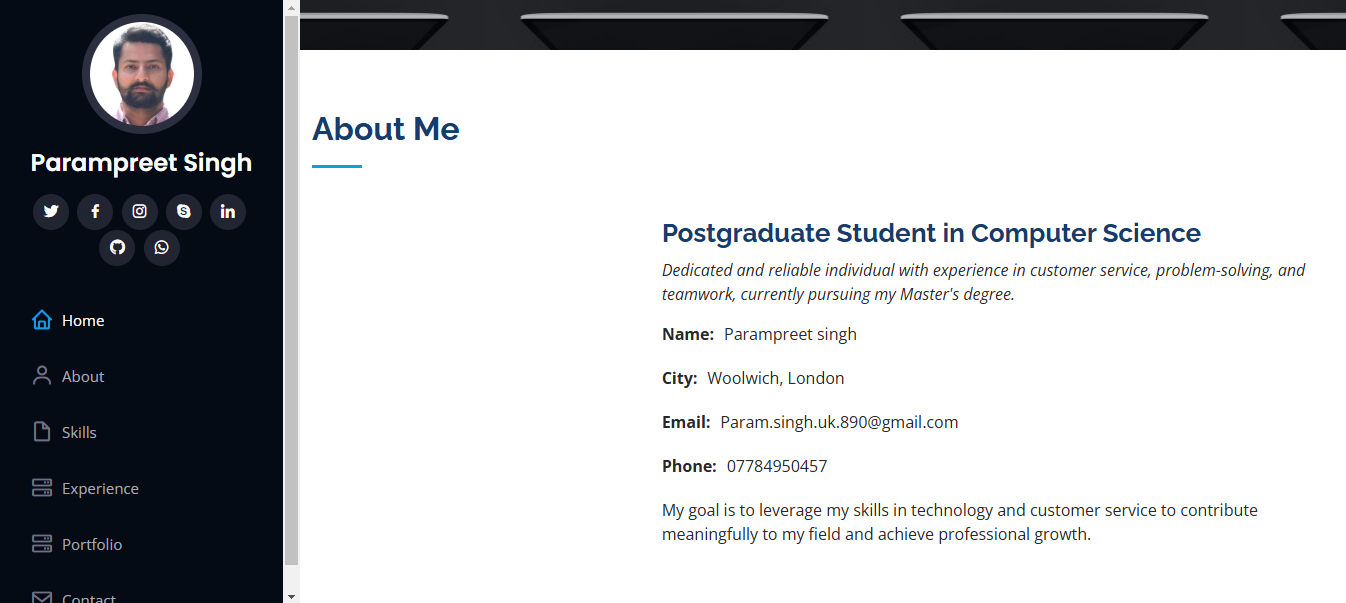
The portfolio website is based to offer a comprehensive review of an individual’s professional historical past and abilties, prepared into awesome pages for smooth navigation (Arun et al., 2022).



**Figure 1: Home Page**

(Source: Self-created)

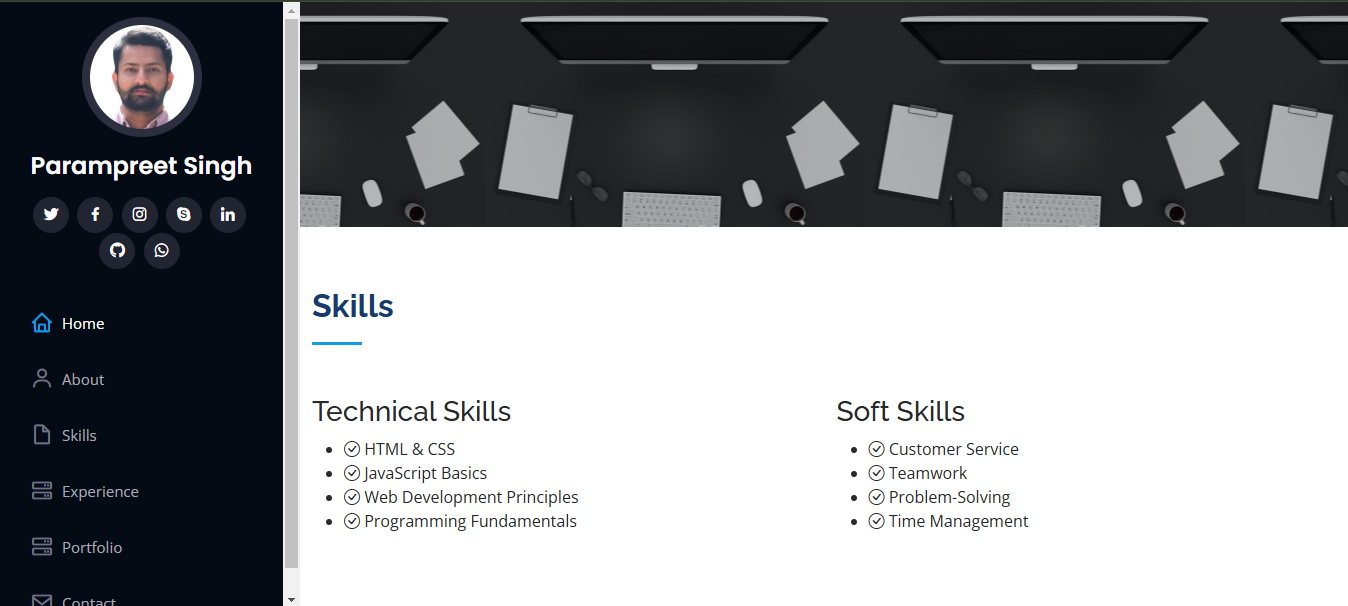
**Home Page:** This page serves as the first factor of contact, imparting an overview of the website. It includes a short creation, profession goal, and highlights the individual’s professional desires and aspirations.



**Figure 2: About Page**

(Source: Self-created)

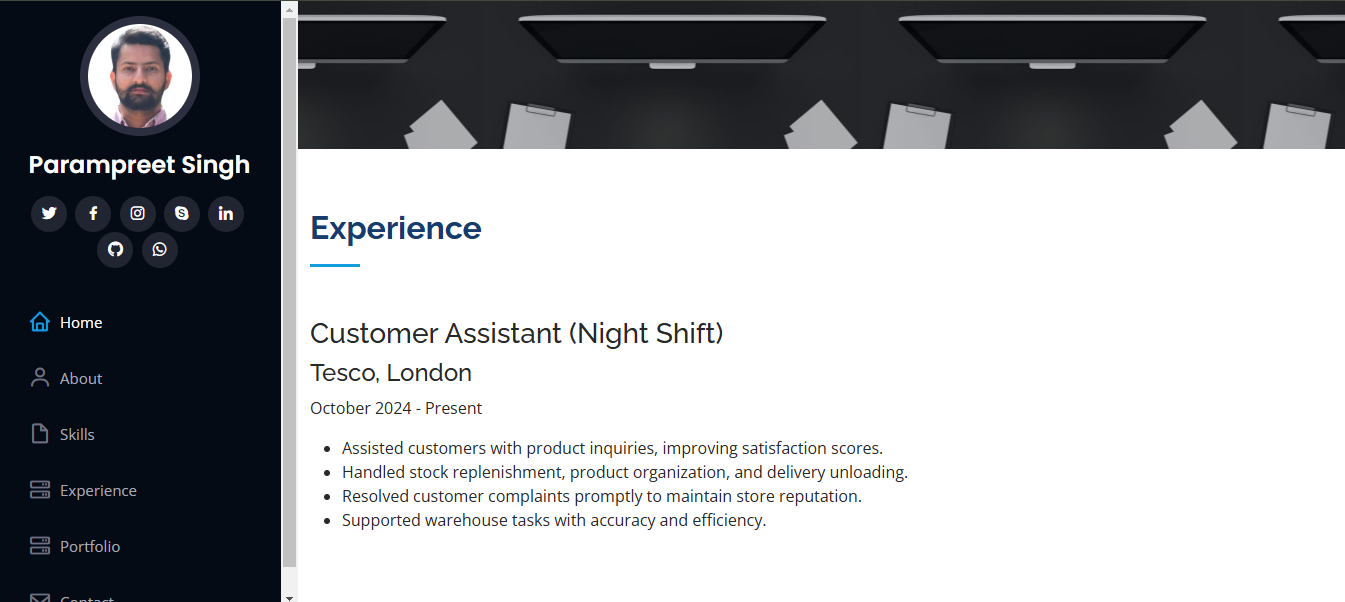
**About Page:** This segment provides specified records about the man or woman, consisting of instructional heritage, profession journey, and destiny aspirations. It lets in site visitors to benefit a deeper understanding of the man or woman at the back of the portfolio.



**Figure 3: Skills Page**

(Source: Self-created)

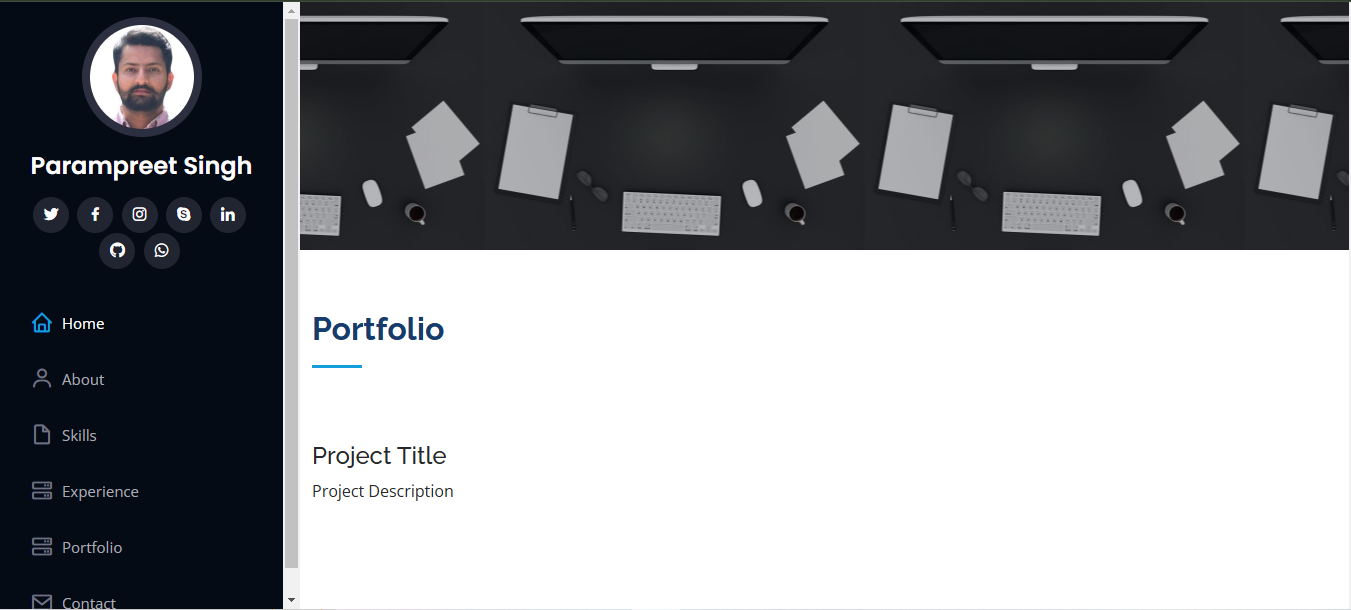
**Skills Page:** A committed segment to showcase each technical and soft abilities. This web page makes use of visible factors, such as graphs and charts, to effectively constitute the user’s talent in various areas, including programming, hassle-fixing, and conversation.



**Figure 4: Experience Page**

(Source: Self-created)

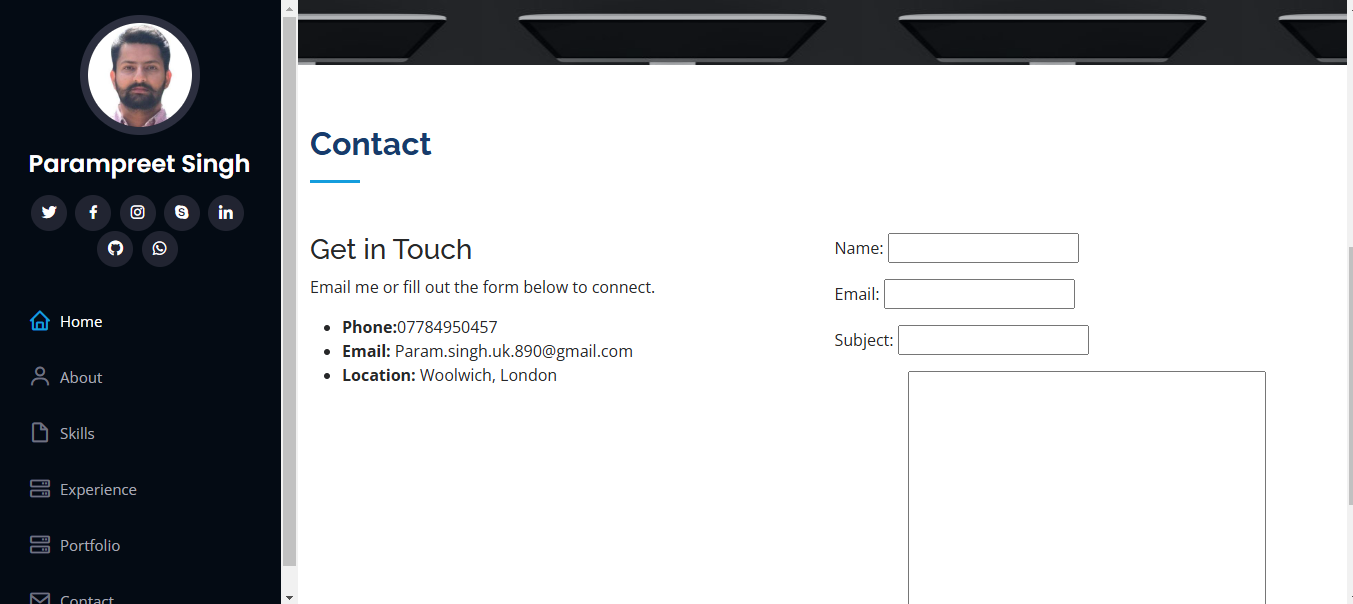
**Experience Page:** This web page summarizes the person’s work revel in, detailing key roles, responsibilities, and achievements. It gives insights into how past stories make contributions to their cutting-edge ability set.



**Figure 5: Portfolio Page**

(Source: Self-created)

**Portfolio Page:** A gallery of finished projects, showcasing real-global packages and achievements. Each assignment includes descriptions and hyperlinks to similarly information.



**Figure 6: Contact Page**

(Source: Self-created)

**Contact Page:** A touch shape included with Django, allowing site visitors to easily get in touch. Form submissions are stored in a MySQL database, ensuring efficient record control and interplay tracking.

## Key Features of the Website:

The portfolio website functions with a **responsive design**, making sure it's miles completely optimized for cell and tablet devices. **Interactive elements** include the use of JavaScript, which includes dynamic shape submissions and easy animations that enhance personal engagement. The internet site additionally utilizes **graphical images and graphs** to visually represent competencies, revel in, and projects. Project pics provide a clear show of finished paintings, even as ability and enjoy graphs offer a concise, visual illustration of skillability and achievements. These functions together create an engaging, user-pleasant experience, highlighting the man or woman’s professional competencies in a cutting-edge, visually attractive layout (Odeniran, 2023).

# Backend Development with Django

## Django Setup:

Django became used to set up the backend of the portfolio internet site, permitting efficient management of information. **Models** were created to shop important statistics, including portfolio assignment details and phone shape submissions, ensuring statistics' staying power. The **views** perspectives have been designed to deal with requests, retrieve statistics from the database, and pass it to the templates for rendering dynamic content material on the front end. **Templates** had been used to structure HTML pages, showing the facts in a smooth and organized manner. Django’s MVC architecture facilitated the seamless integration of backend capability with the user interface, enhancing the website’s interactivity and overall performance (Martins, 2021).

## Database Integration (MySQL):

The **MySQL database** became integrated with Django by means of configuring the DATABASES, putting inside the settings record, specifying MySQL as the database engine and providing the vital credentials, consisting of the database call, consumer, and password. To shop for applicable information, **models** have been created for each portfolio task and contact shape submission. For instance, a Project model shops mission details like name, description, and pics, while a ContactMessage model stores contact form entries, together with calls, electronic mail, and messages.

After defining the fashions, **migrations** were implemented the use of Django's make migrations and migrate commands to generate and observe the database schema.

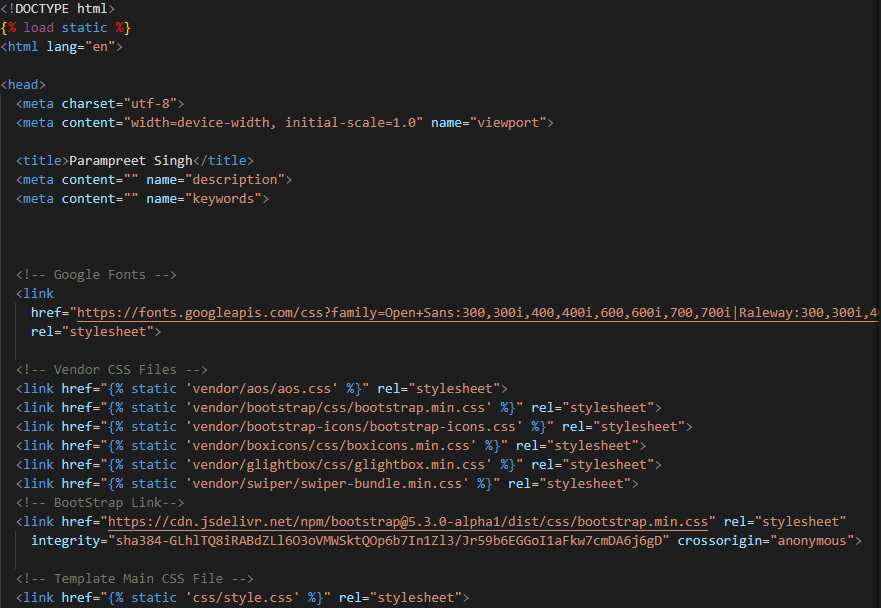
To make certain facts integrity, **form validation** become applied in Django. Before saving statistics from the contact form to the database, the form assessments for required fields, right electronic mail formatting, and other validation policies. This ensures that best legitimate and complete statistics is stored in the MySQL database.

## Django Views and Templates:

In Django, **views** are responsible for fetching statistics from the database and passing it to templates for rendering. For instance, portfolio initiatives and abilities information are retrieved from the database via model queries within the perspectives. These views then send the information to the corresponding **templates**, where dynamic content is displayed. Django’s **template language** allows for the smooth integration of variables, loops, and conditionals to render information dynamically. This permits the internet site to reveal up-to-date portfolio tasks, talents, and other personalised content material. Templates make sure that the content material is offered in a well-structured and visually appealing manner.

# Frontend Development with HTML and CSS

## HTML Structure:

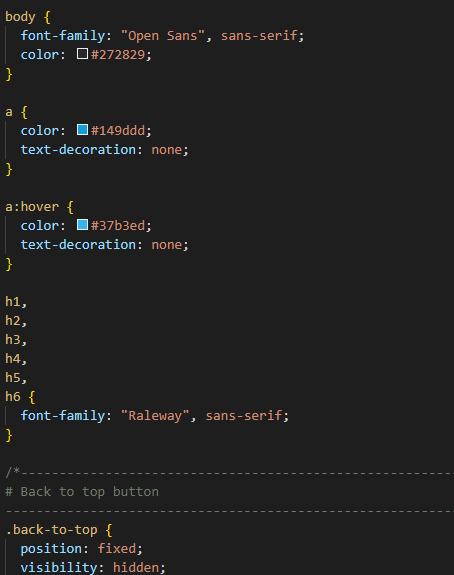


**Figure 7: HTML Code**

(Source: Self-created)

The HTML structure of the internet site is organized into distinct sections, every corresponding to a distinct web page: **Home, About, Skills, Experience, Portfolio,** and **Contact**. Each page is dependent the use of semantic HTML tags along with <header>, <footer>, <section>, <article>, and <nav>, which beautify each accessibility and SEO. These tags make the content material easily comprehensible for display readers and search engines like Google. For instance, the **Home Page** makes use of <header> for the principal identity, and the **Portfolio Page** makes use of <article> for individual tasks. Proper use of those tags improves consumer experience and seek engine ranking (Lama, 2024).

## CSS for Styling:



**Figure 8: CSS Code**

(Source: Self-created)

The CSS for the portfolio website turned into a **responsive design** in thought, ensuring it adapts seamlessly to numerous screen sizes, from computers to cell devices. Modern CSS techniques like **Flexbox** and **Grid Layout** have been used to create a flexible, organized layout. **Media queries** had been incorporated to adjust the layout for exceptional breakpoints, enhancing consumer experience on all devices. For **visible aesthetics**, an expert colour scheme, contemporary fonts, and steady spacing have been selected to maintain an easy and polished appearance. The layout and styling choices follow present-day internet layout trends, ensuring the website is both functional and visually appealing.

## 3D and Dynamic Effects:

3D effects were carried out on shape **fields, buttons**, and **page transitions**, the use of CSS to create a more interactive and attractive consumer reveal. For instance, shape fields and buttons raise with diffused **box-shadow** and **translateY** effects whilst targeted or hovered, giving them a 3-D look. **Hover effects** on buttons and hyperlinks also create a dynamic feel, with easy transitions and scaling results. Additionally, **scrolling animations** were used to vanish in factors as users scroll through the page, maintaining the experience energy. These animations and consequences upload intensity, interactivity, and varnish, enhancing each usability and visual attraction.

# JavaScript for Dynamic and Interactive Elements



**Figure 9: Java Script**

(Source: Self-created)

## Form Validation:

JavaScript changed into used to implement **real-time form validation** at the touch form, ensuring customers publish accurate and whole information. As users fill out the form, JavaScript exams each subject dynamically for mistakes, along with missing or invalid entries. For instance, the email discipline is validated for correct electronic mail format, and the message field guarantees text is entered. Error messages are displayed right away if validation fails, guiding customers to correct their enter. This technique enables improve records accuracy and prevents incomplete or incorrect submissions, improving consumer enjoy and making sure reliable statistics collection (Koltraka, 2024).

## Interactive Features:

The portfolio website contains **interactive features** to enhance user engagement. For displaying **skills and experience** in records, JavaScript libraries like **main.js** have been used to create dynamic, visually appealing graphs. These graphs represent skills skillability or enjoy tiers and are up to date in actual-time based totally on facts from the backend. Custom JavaScript scripts fetch this facts and render the graphs dynamically, offering an interactive and informative experience for the consumer.

## Enhancing User Experience:

To enhance the overall person, enjoy, the portfolio internet site integrates **smooth scrolling navigation** and **JavaScript animations**. The smooth scrolling feature guarantees a fluid transition among sections when users click on on the navigation hyperlinks. Instead of jumping from one segment to any other, the page smoothly scrolls, growing a more polished and seamless surfing experience. This characteristic improves the drift of interplay and makes navigating the website feel greater intuitive (O'Reilly,2021).

Additionally, **JavaScript animations** are used to make the website extra dynamic and tasty. For instance, lively text is displayed in the **hero segment**, where key phrases or titles seem with a fade-in impact, grabbing customers' interest straight away. Another animation effect consists of **fading in images** as customers scroll down the web page. As they reach different sections, photographs and content gradually appear, making the website feel alive and interactive.

These upgrades, consisting of easy scrolling and JavaScript animations, make contributions to a greater attractive and responsive revel in for customers, encouraging them to explore the internet site further.

# Database Integration and MySQL

## Database Setup:

The **MySQL database** is configured in Django with the aid of specifying the database engine, call, consumer, and password inside the settings.Py record. A MySQL database change was created with appropriate **user permissions** to grant admission. Connection settings have been well configured, permitting Django to interact with MySQL for records garage (Ho, 2024).

## Data Storage:

Data from the **contact form** is saved in the MySQL database through Django models. The touch form entries are stored in a ContactMessage model, securely stored to the database.

## Querying and Displaying Data:

Django’s **ORM** is used to question and retrieve dynamic statistics, inclusive of **form submissions**. The ORM lets in clean filtering and sorting of data saved in MySQL. Retrieved statistics is handed to Django templates for dynamic display, making sure up-to-date content on the internet site.

## Handling User Submissions:

Contact form submissions are saved inside the database thru Django’s **ContactMessage** model. Data is verified to ensure correctness, consisting of e-mail formatting and required fields. Upon a success submission, customers get hold of a confirmation message, and their info are saved securely inside the MySQL database for future reference (Rinaldi, 2023).

# Challenges and Solutions

**Integrating JavaScript with Django**: One of the important thing demanding situations changed into making sure seamless interplay among Django’s backend and JavaScript-based frontend elements. For instance, coping with dynamic content material updates, including form submissions or updating graphs with JavaScript, required right integration with Django's templates. Initially, I faced troubles with information not being dynamically up to date because of mismatched facts handling between JavaScript and Django perspectives (Fernandes et al., 2022).

**Solution:** To cope with this, I used **AJAX** to make asynchronous requests among the frontend and backend, taking into consideration actual-time shape submissions and dynamic updates with out reloading the page. This allowed JavaScript to engage efficiently with Django’s perspectives, making sure clean statistics exchange.

**Making the Website Responsive:** Another giant challenge is making sure that the internet site changes into responsive through one-of-a-kind gadgets. Due to varying display screen sizes and resolutions, positive elements appeared misaligned or distorted, mainly on cell gadgets.

**Solution:** I used **CSS media queries** to create responsive layouts that adapt to various display sizes. Additionally, I employed **Flexbox** and **Grid Layout** to create a flexible layout structure that adjusts based totally on the viewport length. This ensured a constant experience across laptop, tablet, and mobile gadgets.

**Setting Up MySQL:** Configuring the **MySQL database** and making sure Django changed into properly linked to it turned into to start with difficult, particularly with dealing with database migrations and person permissions.

**Solution:** By cautiously configuring the **DATABASES** setting in Django's settings.py and verifying MySQL server settings, I successfully set up the database. I additionally created Django models to manipulate contact shape submissions and portfolio records.

**User Experience Challenges**

The **user experience** had to be optimized, especially for quick, smooth navigation and data submission. **JavaScript** animations were used to engage users and to guide users through different content areas, and **smooth scrolling** navigation was used to avoid abrupt jumps between sections.

# Testing and Deployment

**Testing the Website**: Pre-launch testing was also completed to make certain that the website operates properly on any **device** and in any **web browser**. The website was tested on desktop, tablet, and mobile devices to ensure **responsiveness** and optimal viewing experience. A cross-browser compatibility check was carried out on Chrome, Firefox, Safari, and Edge to provide the same level of performance. Particular emphasis was placed on testing the **contact form** feature and describing the submission of the form, validation of the data being sent to the server, and proper data storage in the **MySQL** database (Bedi, Begam & Godara, 2024).

**Performance Optimization:** To improve the performance of the website they incorporated different measures of optimization. **Image optimisation** was performed to increase the size of the images without intents to decrease quality to make the pages load faster. **JavaScript minification** was implemented to decrease the file size and improve page load speed. As for images and other objects, methods of **lazy loading** were applied: such elements are loaded only when the user interacts with the content of the page.

**Deployment:** The website was deployed on **Heroku** which is one of the cloud application hosting services. In addition to using Heroku to set up the environment and database for the application, **MySQL** was ready for production use with some additional add-ons also installed and the right connection details included in the settings.py file. The actual deployment was performed to ensure the website, which has been developed, works fine with the live server.

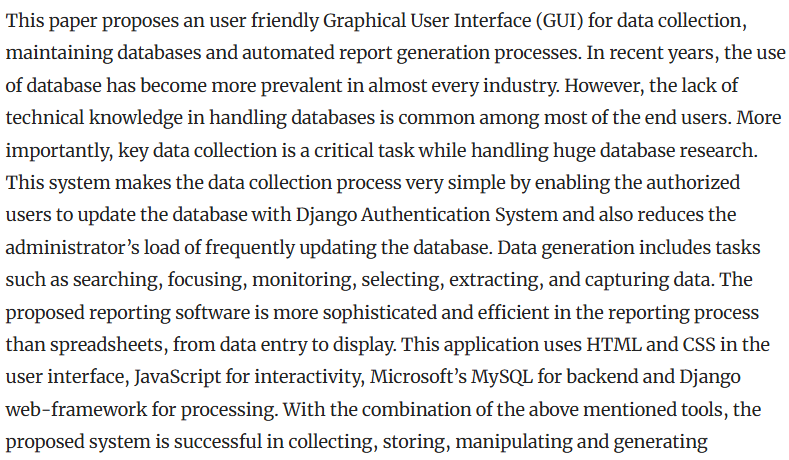
# Conclusion

In summary, the portfolio website serves the purpose of presenting own skills, work experience and projects as well as offering an entertaining design with nice animations. The website was coded using **Django** for back-end programming, **HTML** and **CSS** for web designing and J**avaScript** for interaction and **MYSQ**L for the Database. All of these technologies were well integrated to build a certainly active, sensible and moreover user-friendly platform.

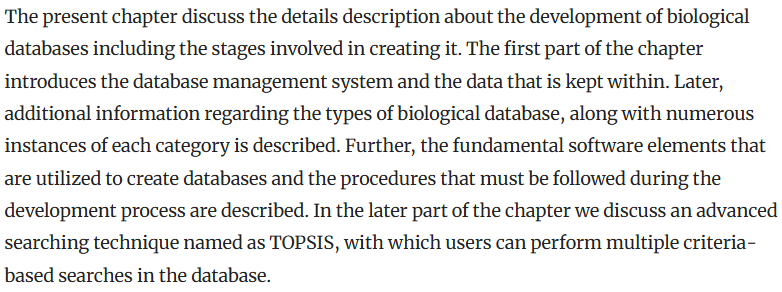
It also helped me expand my web development skills in using Django and MySQL for backend administration and also **HTML CSS** and **JavaScript** for the frontend side of the project. Blending the dynamics into the design and making sure that the transitions as far as the user interactions are considered was both fun and difficult simultaneously.

# References

Arun, C., Priyenka, S., Sagarika, L., & Sneha, S. (2022). A Web-Based Application for Data Collection and Report Generation Using Django. In *ICT Systems and Sustainability: Proceedings of ICT4SD 2021, Volume 1* (pp. 519-527). Springer Singapore. Retrieved from: <https://www.researchgate.net/profile/Mahtab-Shahin/publication/357566347_Implementation_of_e-Birth_Registration_Systems_Potential_and_Challenges_The_Case_Study_of_Iran/links/61dd99e73a192d2c8af31e7b/Implementation-of-e-Birth-Registration-Systems-Potential-and-Challenges-The-Case-Study-of-Iran.pdf#page=525> [Retrieved on22.01.25]



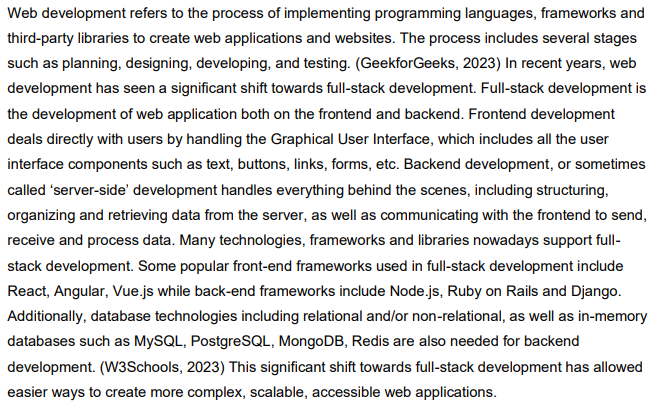
Bedi, J., Begam, S., & Godara, S. (2024). Development of Biological Databases for Genomic Research. In *Genomics Data Analysis for Crop Improvement* (pp. 309-323). Singapore: Springer Nature Singapore. Retrieved from: <https://link.springer.com/protocol/10.1007/978-981-99-6913-5_12> [Retrieved on: 22.01.25]



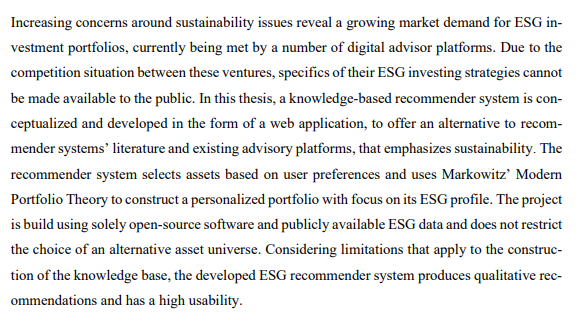
Fernandes, E. C., Camatti, J. A., Tabor, S. J. F., Romanel, L. G., Brown, L., & Borsato, M. (2022, November). Value Stream Mapping Application Enabled by Django Web Framework. In *International Conference on Production Research* (pp. 700-707). Cham: Springer Nature Switzerland. Retrieved from: <https://www.researchgate.net/profile/Milton-Borsato/publication/372665987_Value_Stream_Mapping_Application_Enabled_by_Django_Web_Framework/links/653a6b8c5d51a8012b715392/Value-Stream-Mapping-Application-Enabled-by-Django-Web-Framework.pdf> [Retrieved on: 22.01.25]



Ho, H. (2024). Developing a full-stack E-commerce application with Next. js, JavaScript, React and MongoDB. Retrieved from: <https://www.theseus.fi/bitstream/handle/10024/819599/Hoan_Ho_Developing_a_fullstack_Ecommerce_application_with_Next.js_JavaScript_React_and_MongoDB.pdf?sequence=2> [Retrieved on: 22.01.25]



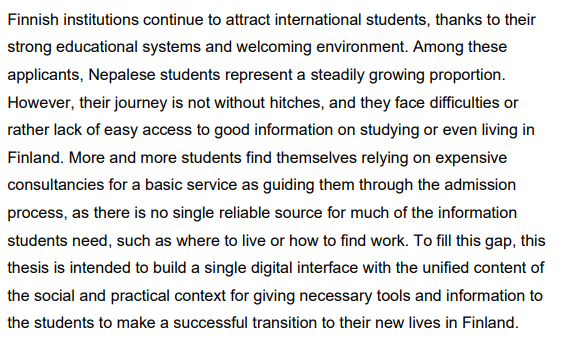
Koltraka, E. (2024). *ESG investing: Developing a recommender system for portfolio construction* (Doctoral dissertation, Hochschule für Angewandte Wissenschaften Hamburg). Retrieved from: <https://reposit.haw-hamburg.de/bitstream/20.500.12738/16715/1/BA_ESG%20investing.%20Developing%20a%20recommender%20system%20for%20portfolio%20construction_geschw%C3%A4rzt.pdf> [Retrieved on: 22.01.25]



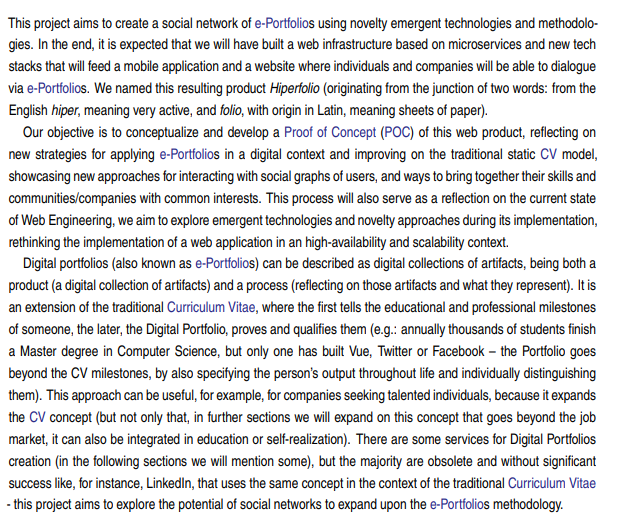
Lama, U. (2024). News portal for Nepalese students applying to study in Finland (WordPress): Providing students with information and resources. Retrieved from:

<https://www.theseus.fi/bitstream/handle/10024/877278/Lama_Ugyen.pdf?sequence=2>

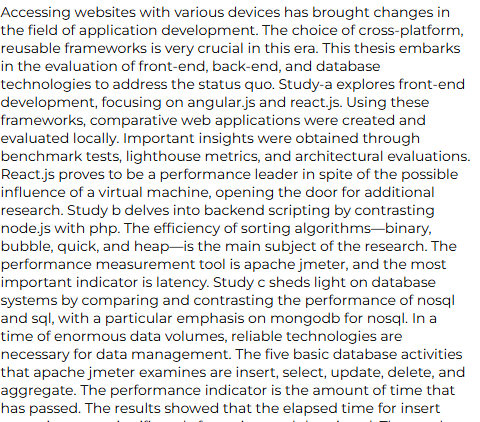
[Retrieved on: 22.01.25]



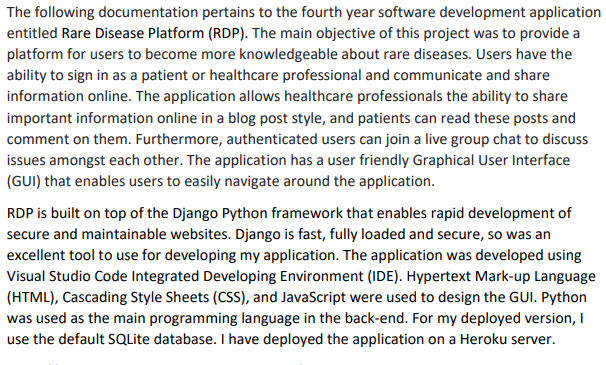
Martins, P. J. P. (2021). Development of an e-portfolio social network using emerging web technologies. Retrieved from: <https://core.ac.uk/download/pdf/588865803.pdf> [Retrieved on: 22.01.25]



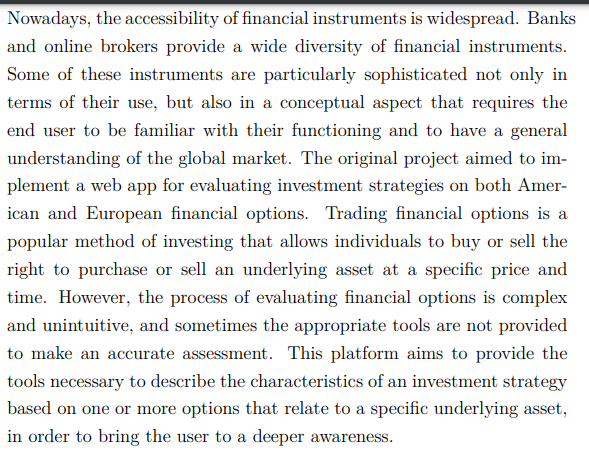
Odeniran, Q. (2023). Comparative Analysis of Fullstack Development Technologies: Frontend, Backend and Database. Retrieved from: <https://digitalcommons.georgiasouthern.edu/cgi/viewcontent.cgi?article=3883&context=etd> [Retrieved on: 22.01.25]



O'Reilly, C. (2021). *Rare Disease Platform: Technical Report* (Doctoral dissertation, Dublin, National College of Ireland). Retrieved from: <https://norma.ncirl.ie/5024/1/conororeilly.pdf> [Retrieved on: 22.01.25]



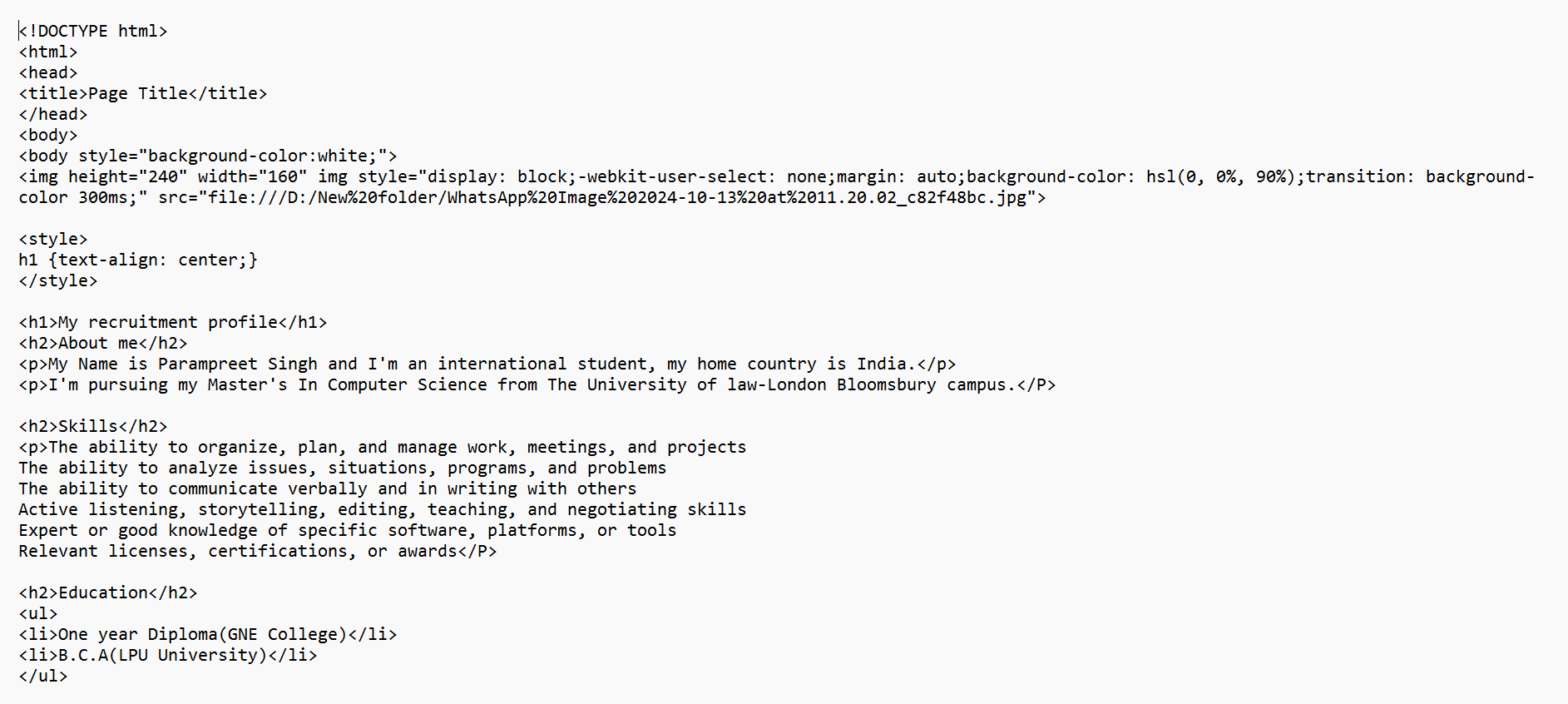
Rinaldi, R. (2023). *Study and functional improvements development for an Options Strategy Builder Platform* (Doctoral dissertation, Politecnico di Torino). Retrieved from: <https://webthesis.biblio.polito.it/secure/27677/1/tesi.pdf> [Retrieved on: 22.01.25]



**Consolidate Tasks**

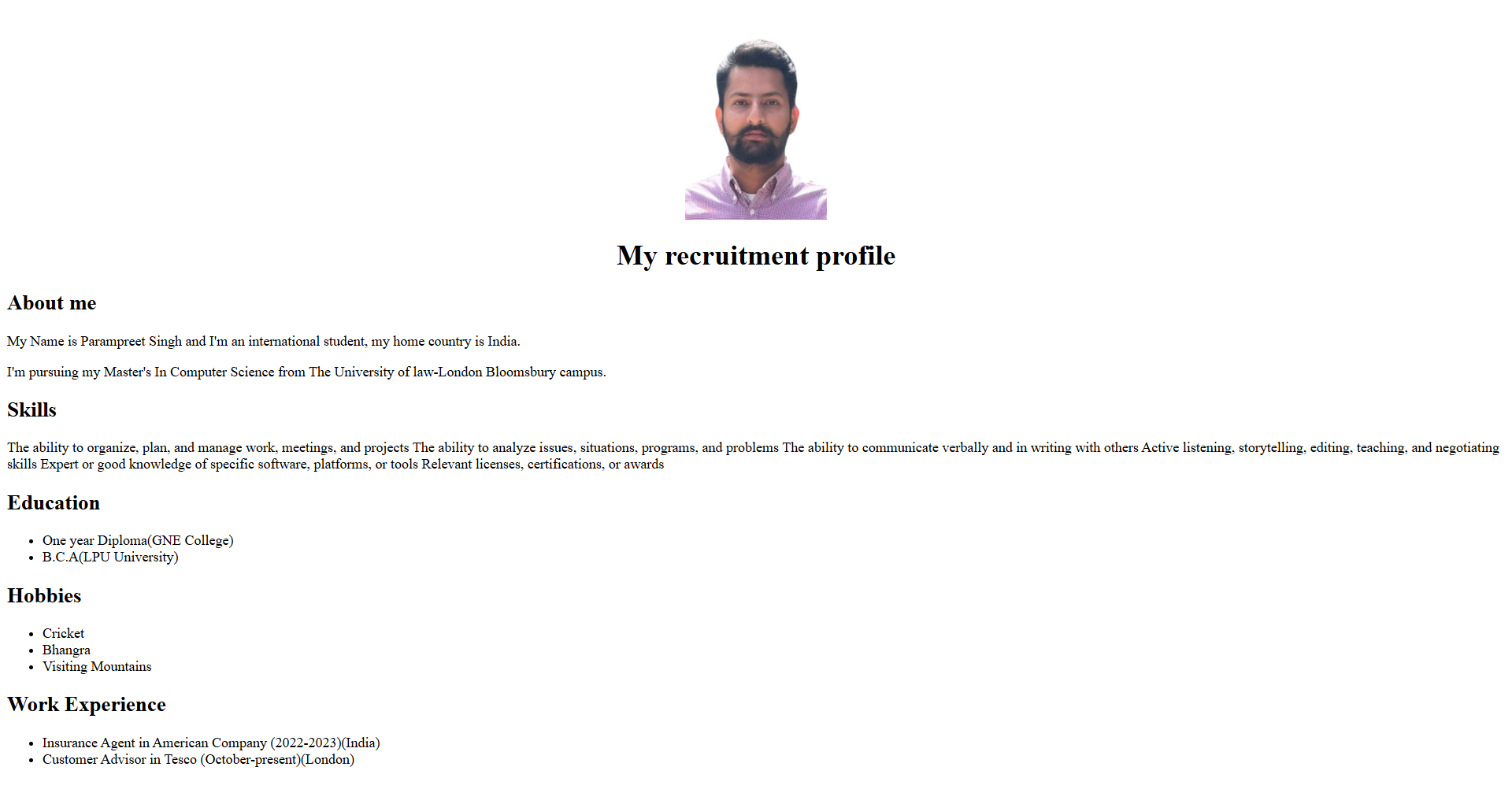
**Task 1**

**Input**

****

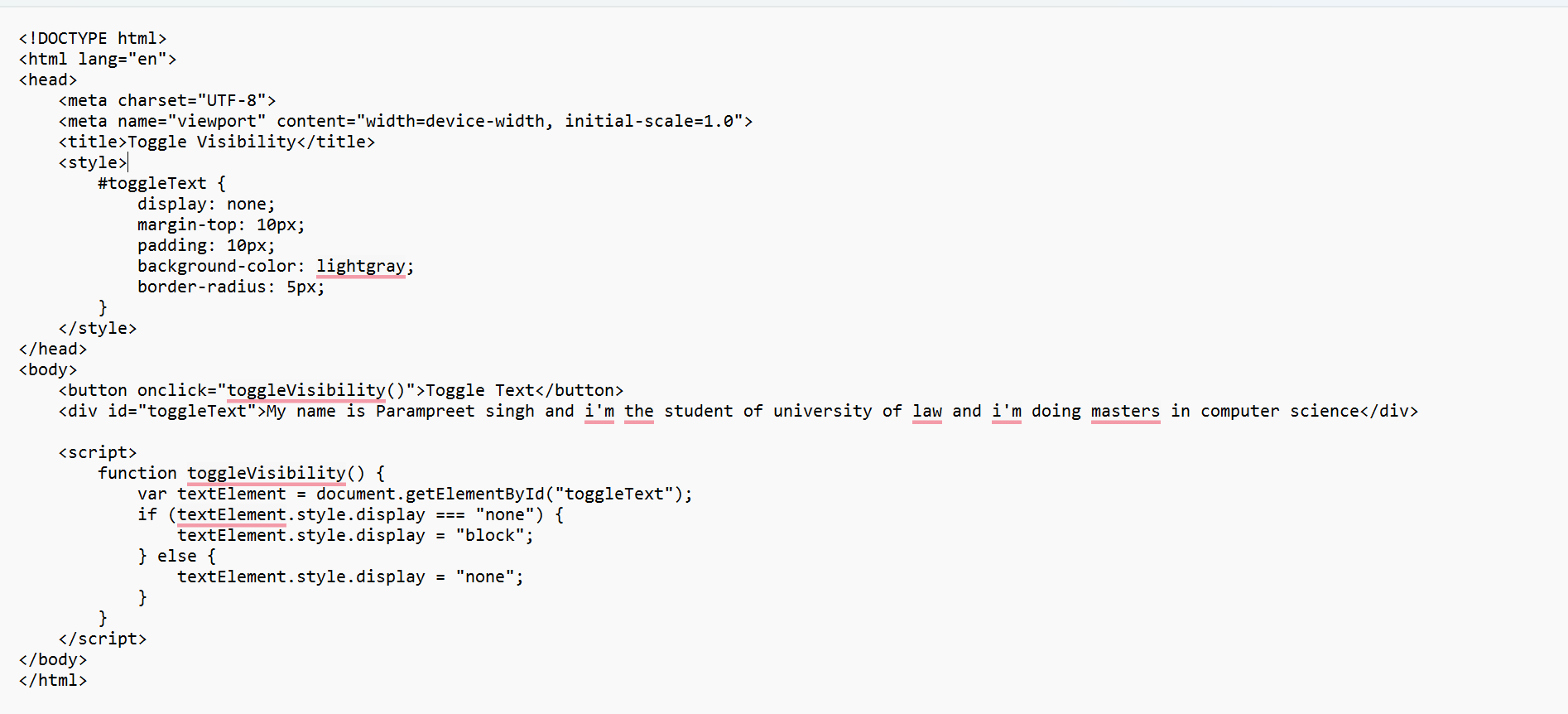
****

**Output**

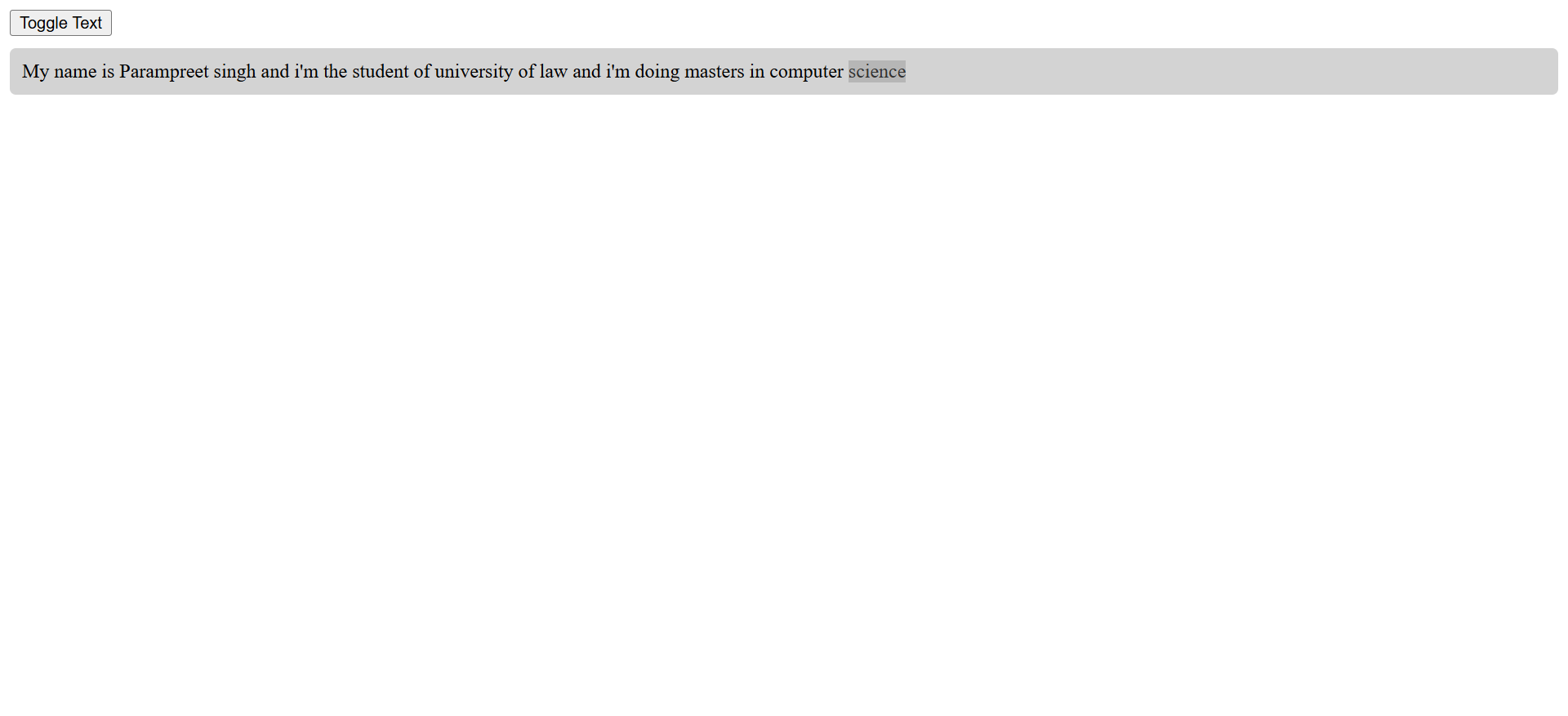


**Task 3**

**Input**



**Output**



**Task 4**

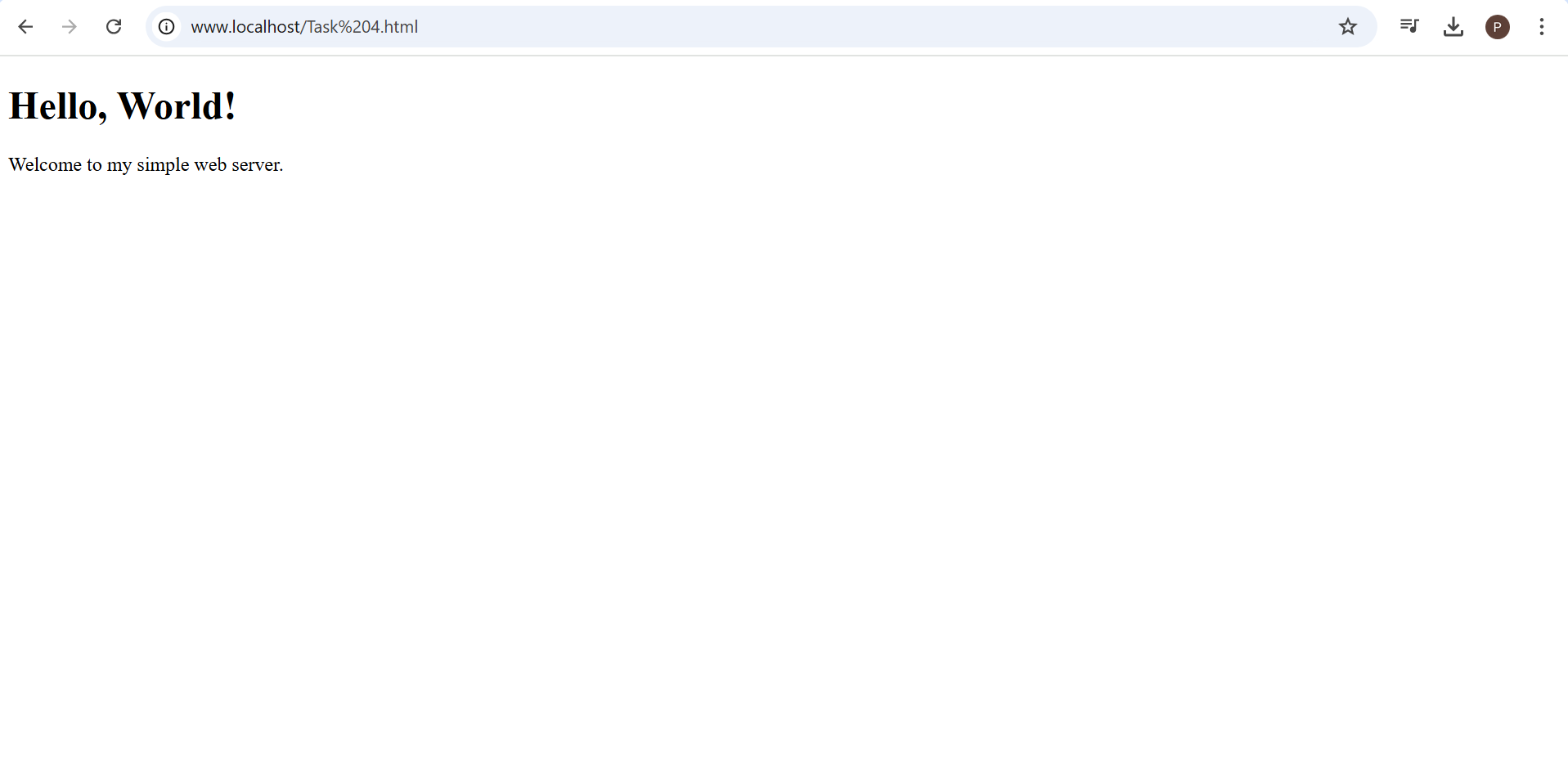
**Input:-**



**Output:-**



**Task 5**



I open my hello world file in local wamp server. Here I attached the screenshot.

**Task 6**



Here, I create and publish my blog post by using Wordpress, which is one of the post popular CMS Platforms and you can also see my blog by using this link which I attach the link <https://raman748.wordpress.com>

**Task 8**

**Input:-**



**Output:-**

<file:///C:/Users/hp/Desktop/web%20technologies/Unit%208/multimedia.html>

****