**Project Report**

## “Bookmark Saver Website ClipLy”

**A Project Proposal**

**By**

**Simrat Kaur**

**Thapar Institute of Engineering and Technology, Patiala**

## Under the guidance of

**Industry Mentor**

Mr. Udayveer Deswal

Drish Infotech.

## DECLARATION

I hereby declare that the presented report entitled **“Bookmark Saver Website ClipLy”** with **Drish Infotech, Chandigarh** has been uniquely prepared by me and it is an original work. This report is being submitted for fulfilling the requirement of B.Tech in Computer Engineering as a Summer Training Project, at **Thapar Institute of Engineering and Technology ,Patiala.**

I also hereby declare that it has not previously formed the basis for the award to me for any degree

/ associateship, fellowship or other similar title of any Institute.

Simrat Kaur

102203201

Place: Patiala

## ACKNOWLEDGEMENT

Summer Internship training is a very crucial part of my Engineering program and I take this as an opportunity to express my heartfelt gratitude to Drish Infotech, Chandigarh which helped me to complete this project. It gives me immense pleasure to present this project report on ― “Bookmark saver Chrome Extension”. It’s never easy to carry out work without the guidance and help of few motivating people in life. I am happy to take this opportunity to express gratitude to those who have been helpful to me in completing this project report.

I am highly indebted to my Industry mentor Mr. Udayveer Deswal for his constant guidance and Mentor ship. I am extremely thankful to Sanjeev Sir and Komal Mam for their constant support and guidance. They have been instrumental in assisting me in my work and encouraging me to – gather information from all sources within my reach.

I am highly indebted to all the employees of Drish Infotech for their time and cooperation they gave from their busy schedule which helped me in completion of this project.

Lastly I would like to mention and thanks my friends, parents and fellow peers who in a way directly or indirectly has contributed for the completion of my project report and encouraged me all throughout the journey.

Thanking you,

Simrat Kaur

102203201

### ABSTRACT

### The rapid evolution of digital content consumption has necessitated the development of tools that help users efficiently manage the vast amount of information encountered daily. ClipLy is a sophisticated bookmark saver Chrome extension designed to streamline the process of saving, organizing, and retrieving online resources. This project report delves into the development, features, and impact of ClipLy, highlighting its significance as an essential tool for modern web users.

### The genesis of ClipLy arose from the need for a user-friendly and efficient solution to manage bookmarks. Traditional bookmarking methods often lack organizational capabilities, leading to cluttered and hard-to-navigate lists. ClipLy addresses these issues by offering a comprehensive platform that not only saves bookmarks but also categorizes them for easy access.

### The core functionality of ClipLy revolves around its seamless integration with the Chrome browser. Users can save any webpage with a single click, and the extension automatically captures the URL, title, and a brief description. This information is then stored in a structured database, enabling users to sort and search their bookmarks effortlessly. The intuitive user interface allows for the creation of custom categories and tags, further enhancing the organizational capabilities of ClipLy.

**TABLE OF CONTENT**

|  |  |
| --- | --- |
| **CHAPTER NAME** | **PAGE NUMBER** |
| **CHAPTER- 1**  **(Introduction)** | **6** |
| **CHAPTER- 2**  **(Project Scope and proposal)** | **10** |
| **CHAPTER- 3**  **(Modules)** | **15** |
| **CHAPTER- 4**  **(Technological Stack)** | **18** |
| **CHAPTER- 5**  **(Results and Discussion)** | **21** |
| **CHAPTER- 6**  **(WorkFlow)** | **26** |
| **CHAPTER- 7**  **(Conclusion)** | **44** |

# CHAPTER -1

1.1 INTRODUCTION

Established in 1999, Drish Infotech Limited in Chandigarh has become a leading

name in Computer Training Institutes and IT solutions. Located at Sector 83, Alpha

IT City, Sahibzada Ajit Singh Nagar, Punjab, Drish Infotech boasts over two

decades of experience, delivering innovative technology solutions globally.

Specializing in custom application development, the company creates bespoke

software tailored to clients' needs, from scalable enterprise applications to niche

products. Drish Infotech leverages advanced technologies like AI, machine learning,

IoT, and blockchain to provide comprehensive digital transformation strategies,

enhancing productivity and fostering innovation. Serving industries such as

healthcare, finance, e-commerce, education, and logistics, Drish Infotech is known

for its deep industry insights, agile development methodologies, and commitment to

customer satisfaction. Working on the E-Ticketing website, I gained hands-on

experience in front-end and back-end development using React, Next.js, Node.js,

PostgreSQL, and Apache, enhancing my technical and problem-solving skills.

Drish Infotech specializes in a comprehensive array of services tailored to meet the

unique challenges and opportunities of modern businesses:

1. Custom Application Development: Drish Infotech excels in developing bespoke

software solutions that are meticulously crafted to align with the specific needs and

objectives of its clients. From enterprise-level applications to niche software products,

the company leverages its expertise to deliver scalable, secure and user-friendly

solutions.

1. Digital Transformation Strategies: Recognizing the imperative for businesses to

adapt and thrive in the digital era, Drish Infotech offers robust digital transformation

strategies. These strategies encompass the integration of advanced technologies such

as artificial intelligence, machine learning, Internet of Things (IoT), and blockchain,

empowering organizations to streamline operations, enhance productivity, and drive

innovation.

1. Product Development and Engineering: Drish Infotech is at the forefront of

product development, providing end-to-end services from ideation to deployment.

The company collaborates closely with clients to conceptualize, design, prototype,

and launch innovative products that resonate in the market. This includes

comprehensive testing, iterative development, and continuous improvement to ensure

product excellence.

1. Outsourcing and Managed Services: With a focus on operational efficiency

and cost-effectiveness, Drish Infotech offers reliable outsourcing solutions. These

solutions range from dedicated offshore development teams and IT support to

managed services and maintenance. By outsourcing non-core functions, businesses

can leverage Drish Infotech's expertise to optimize resources and achieve strategic

objectives.

1.2 Company Profile

About Drish Infotech Pvt Ltd:

Drish Infotech Pvt Ltd is a software development company headquartered in Mohali,

Chandigarh. Established the company focuses on delivering innovative IT solutions

to clients across various industries. Drish Infotech specializes in custom software

development, web and mobile application development, e-commerce solutions,

digital marketing, and IT consulting services. The company's mission is to leverage

cutting-edge technology to help businesses achieve their goals efficiently and

effectively.

# CHAPTER -2

**CHAPTER 2: PROJECT SCOPE AND PROPOSAL**

**Project Title: ClipLy Bookmark Saver website**

**Platform/Languages Used:** Python, Django, JavaScript  
**Database:** SQL Lite  
**Scripting Language / Designing:** HTML, CSS, JavaScript

**2.1 Project Scope Definition**

ClipLy is a Chrome extension designed to enable users to save, organize, and manage bookmarks efficiently. Below are the detailed descriptions of the main functionalities:

**2.1.1 User Registration and Login**

* **User Registration:** Users can sign up by providing essential details such as name, email, and password. The system ensures secure password encryption and validation.
* **User Login:** Registered users can log in using their credentials. The system supports session management to keep users logged in securely.
* **Profile Management:** Users can update their profiles, including personal information and password changes.
  + 1. **Bookmark Management**
* **Bookmark Saving:** Users can save any webpage with a single click, capturing the URL, title, and a brief description.
* **Bookmark Editing:** Users can edit bookmark details post-creation to accommodate any changes.
  1. **Admin Control Panel**
* **Dashboard:** A centralized dashboard for admins to view site activity, including user registrations and bookmark activity.
* **User Management:** Tools for admins to view, edit, or deactivate user accounts.
  1. **Reporting and Analytics**
* **Customizable Reports:** Ability for admins to generate custom reports based on specific criteria and export data for further analysis.

**2.5 Project Deliverables**

**Phase 1: Development**

1. **Functional Specifications Document (FSD):** Detailed documentation of functionalities, features, and user stories.
2. **User Interface (UI) Mock-ups/Prototypes:** Visual representations of the application's layout and user flow.
3. **Admin Panel Development:** Dedicated control panel for admins to manage bookmarks, users, and analytics.
4. **Bookmark Management Module:** Interface for saving, editing, and organizing bookmarks.
5. **User Module:** Interface for users to register, log in, and manage bookmarks.
6. **Quality Assurance (QA) Testing:** Rigorous testing to ensure functionality and fix bugs.

**Phase 2: Deployment and Training**

1. **Deployment Package:** Files and instructions for installing and configuring the application on the client's server.
2. **User Manuals and Training Materials:** Guides and materials for end-users and admins.

**Phase 3: Post-Deployment Support**

1. **Knowledge Base:** Repository of solutions to common issues and FAQs.
2. **Ongoing Bug Fixes and Maintenance:** Plan for addressing bugs and providing updates.
3. **Support Channels:** Dedicated support channels for user assistance.

**2.6 Details in Work**

The project involved several milestones, each focusing on different aspects of the ClipLy system. From learning basic web development skills to deploying the final application, each stage contributed to the project's overall success. The work included:

* **Learning HTML, CSS, and JavaScript:** Building foundational knowledge.
* **Understanding Django:** Gaining insights into modern web frameworks.
* **Creating Login and Admin Functions:** Developing secure user authentication and admin functionalities.
* **Building Support and End-User Modules:** Ensuring comprehensive functionality for support teams and end-users.

**2.7 Specification**

The ClipLy system was built using the following technologies: **Front-End:** HTML, CSS, JavaScript  
**Back-End:** Python, Django

**2.8 Details of Measurement**

The performance and functionality of the system were measured through various testing methods, including functional testing to ensure all features worked correctly, and performance testing to evaluate the system's ability to handle high traffic. Additionally, security measures such as data encryption and regular security audits were implemented to ensure data integrity and user safety.

This detailed project scope ensures a comprehensive and user-friendly ClipLy extension, addressing all critical aspects of bookmark management and user interaction.

# CHAPTER -3

**CHAPTER 3: PROJECT MODULES**

The ClipLy Bookmark Saver Chrome Extension project was divided into several key modules, each focusing on specific functionalities essential for the seamless operation of the system. Below are detailed explanations of each module:

**3.1 User Management**

* **Registration:**
  + **User Sign-Up:** The system allows new users to register by providing necessary details such as name, email address, and password.
* **Login:**
  + **User Authentication:** Registered users can log in using their email and password. The system supports secure session management to maintain user authentication throughout their interaction with the extension.
* **Profile Management:**
  + **User Profile:** Users can view and update their personal information, including name, contact details, and password.

**3.2 Bookmark Management**

* **Bookmark Saving:**
  + **Quick Save:** Users can save bookmarks with a single click, capturing the URL, title, and a brief description.
* **Bookmark Editing:**
  + **Modify Details:** Users can edit bookmark details post-creation, including updating the title, description, and tags.
* **Bookmark Viewing:**
  + **Organized Display:** Bookmarks are displayed in an organized manner, allowing users to filter and search through their saved items easily.

**3.3 Admin Control Panel**

* **Dashboard:** A centralized dashboard for admins to view site activity, including user registrations and bookmark activity.
* **User Management:** Tools for admins to view, edit, or deactivate user accounts.
* **Bookmark Management:** Comprehensive tools for managing bookmarks, categories, and tags.

These modules work together to create a comprehensive and efficient ClipLy Bookmark Saver Chrome Extension, providing a seamless experience for users and administrators alike. Each module is designed to ensure the system's robustness, user-friendliness, and security, ultimately enhancing the overall user experience and operational efficiency.

# CHAPTER -4

**CHAPTER 4**

**TECHNOLOGY STACK**

**4.1 Front-End**

**Technologies Used:**

* **HTML/CSS:** Fundamental technologies for creating and styling web pages, ensuring a well-structured and visually appealing interface.
* **JavaScript:** Enhances interactivity and functionality on the client side.
* **Bootstrap:** A popular front-end framework for responsive design, providing pre-designed components and styles to improve the user interface.

**4.1.1 Front-End Breakdown**

* **User Interface:**
  + **Design:** The user interface (UI) is crafted to be intuitive and user-friendly, prioritizing ease of navigation and a pleasant experience for users.
  + **Responsive Design:** The UI adjusts to various screen sizes and devices, ensuring a seamless experience across desktops, tablets, and mobile devices.
  + **Component-Based Architecture:** Utilizing Bootstrap's components for reusable and maintainable UI elements.

**4.2 Back-End**

**Technologies Used:**

* **Django:** A high-level Python web framework that promotes rapid development and clean, pragmatic design.
* **Python:**  In my bookmark saver app, Python is utilized for backend development with Django, handling user authentication, database interactions, and logic for saving and organizing bookmarks efficiently.
* **PostgreSQL:** A powerful, open-source relational database system known for its robustness and SQL compliance.

**4.2.1 Back-End Breakdown**

* **Data Storage:**
  + **SQL Lite :** Handles the storage of user profiles, event information, and ticket details securely and efficiently.
  + **Data Integrity:** Maintains data integrity through well-defined schemas, constraints, and transactions.
  + **Backup and Recovery:** Regular backups and recovery procedures are implemented to safeguard against data loss.
  1. **Web Server**

**4.3.1 Web Server Breakdown**

* **Serving Static and Dynamic Content:** Efficiently serves static files (HTML, CSS, images) and dynamic content generated by the Django application.
* **Logging and Monitoring:** Sets up comprehensive logging and monitoring to track server performance, detect issues, and ensure smooth operation.

**By utilizing this tech stack, the Django-based application is designed to be robust, secure, and scalable, offering a seamless and efficient experience for both users and administrators. The integration of these technologies ensures that the system can handle complex operations, maintain data integrity, and provide a user-friendly interface.**

# CHAPTER -5

**CHAPTER 5**

**RESULTS AND DISCUSSION**

**5.1 Introduction**

This chapter presents the outcomes of the Cliply web extension development project, which I undertook during my internship at Drish Infotech Pvt. Ltd. in Mohali, Chandigarh. The project involved creating a fully functional web extension using Django and integrating it with a user-friendly front-end interface. The implementation was broken down into several key components: the HTML structure for displaying and managing bookmarks, the CSS styling for a polished and responsive design, and the Django back-end for handling data storage and user interactions. The results reflect the successful integration of these components, demonstrating the effectiveness of the Cliply extension in managing and organizing bookmarks efficiently.

* 1. **Project Overview**

The Cliply web extension was designed to streamline the process of saving and managing bookmarks directly from the browser. Users can save URLs, categorize them, and access them easily through the extension’s interface. The project aimed to address common challenges associated with bookmark management, such as accessibility, organization, and retrieval efficiency.

* 1. **Development Process**

**5.3.1 Planning and Requirements Gathering** The development process began with a detailed planning phase. Key requirements were identified, including:

* **Resource gathering for my bookmark saver app involved learning Python extensively, particularly focusing on Django for the backend framework. This included understanding models, views, and templates to manage and display bookmarks effectively. I delved into Django’s authentication system to ensure secure user logins and data privacy. Additionally, I explored Python libraries and tools for web scraping and API interactions to enrich bookmark functionalities. This comprehensive Python learning process empowered me to build a robust, scalable, and user-friendly bookmark saver application.**
* **User Interface (UI) Design:** Creating a visually appealing and intuitive UI that allows users to manage bookmarks effortlessly.
* **Functionality:** Implementing features for saving, organizing, and retrieving bookmarks.
* **Integration:** Ensuring seamless communication between the web extension and the Django back-end.

**5.3.2 Designing the User Interface** The UI design involved creating mockups and wireframes to visualize the extension’s layout and features. The design process focused on:

* **User Experience (UX):** Ensuring that the extension is easy to use and navigate.
* **Responsiveness:** Designing the interface to work well across different devices and screen sizes.
* **Aesthetics:** Using a clean and modern design to enhance user engagement.

**5.3.3 Developing the Front-End** The front-end development involved:

* **HTML:** Structuring the extension’s interface with HTML elements such as forms for adding bookmarks, lists for displaying saved bookmarks, and buttons for interaction.
* **CSS:** Styling the extension to ensure a cohesive and attractive appearance. This included layout adjustments, color schemes, and responsive design considerations.
* **JavaScript:** Adding functionality to the front-end, including event handling for saving and managing bookmarks, and updating the UI dynamically based on user interactions.

**5.3.4 Back-End Development with Django** The back-end development focused on:

* **Django Setup:** Configuring the Django project, including setting up models, views, and templates.
* **Models:** Designing data models to store bookmark information, user details, and other relevant data.
* **Views and Templates:** Implementing views to handle user requests and render templates to display data.
* **APIs:** Creating APIs to enable communication between the web extension and the Django server.
  1. **Results**

**5.4.1 Functionality and Performance** The Cliply extension demonstrated the following results:

* **User Interaction:** Users were able to save, organize, and retrieve bookmarks efficiently through the extension’s interface.
* **Performance:** The extension performed well, with quick response times and minimal lag during bookmark management tasks.
* **Integration:** The seamless integration between the extension and the Django back-end ensured reliable data handling and user experience.

**5.4.2 User Feedback** User feedback was gathered through surveys and direct interactions. Key insights included:

* **Ease of Use:** Users appreciated the intuitive design and ease of managing bookmarks.
* **Functionality:** Positive feedback was received regarding the extension’s features, including bookmark categorization and retrieval.
* **Suggestions for Improvement:** Users provided suggestions for additional features and enhancements, which will be considered for future updates.

**5.4.3 Challenges and Solutions** Several challenges were encountered during development:

* **Integration Issues:** There were initial challenges in ensuring smooth communication between the web extension and Django back-end. These were resolved through debugging and refining the API endpoints.
* **User Interface Adjustments:** Fine-tuning the UI to accommodate various screen sizes and devices required iterative adjustments and testing.
  1. **Discussion**

**5.5.1 Achievements** : The Cliply project successfully achieved its primary objectives:

* **Effective Bookmark Management:** The extension provides users with a robust solution for managing bookmarks, addressing common issues such as organization and accessibility.
* **Seamless Integration:** The integration between the front-end extension and Django back-end was accomplished effectively, ensuring a smooth user experience.

**5.5.2 Lessons Learned** The development process provided several valuable lessons:

* **Importance of User Feedback:** Gathering and incorporating user feedback

was crucial for refining the extension and enhancing its functionality.

* **Testing and Debugging:** Rigorous testing and debugging were essential for identifying and resolving issues, ensuring the extension’s reliability and performance.

**5.5.3 Future Work** Future work for the Cliply project includes:

* **Feature Enhancements:** Adding new features based on user feedback, such as advanced search options and improved categorization.
* **Performance Optimization:** Continuing to optimize the extension’s performance to handle larger volumes of data and ensure responsiveness.
* **Cross-Browser Compatibility:** Expanding compatibility to support additional browsers and platforms.

# CHAPTER -6

**6.2 Project Workflow-**

**6.2.1 Creating the Django Project** The initial setup involved creating a new Django project configured to run on a local development server. This included setting up the project structure and configuring the necessary dependencies.

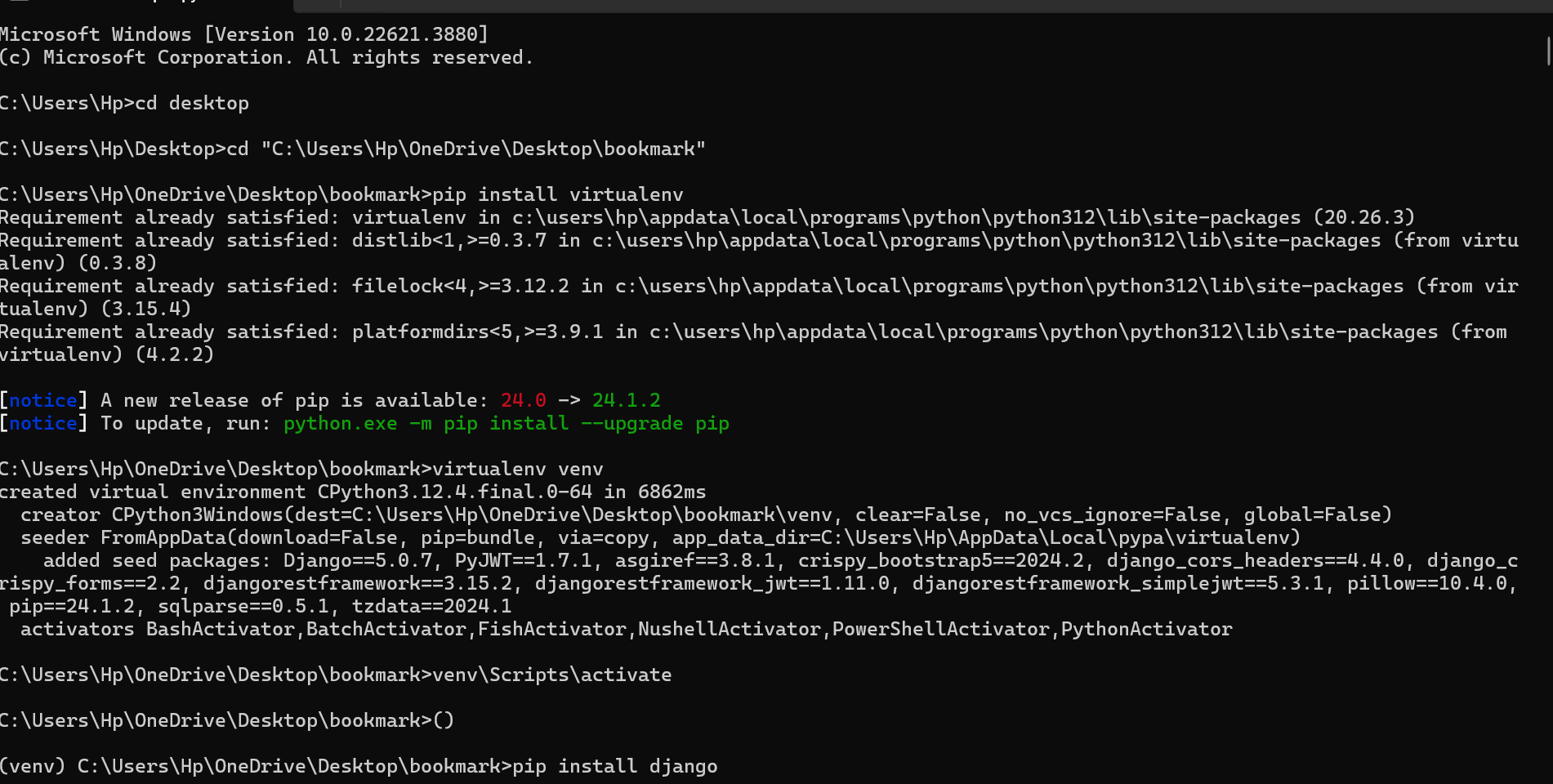
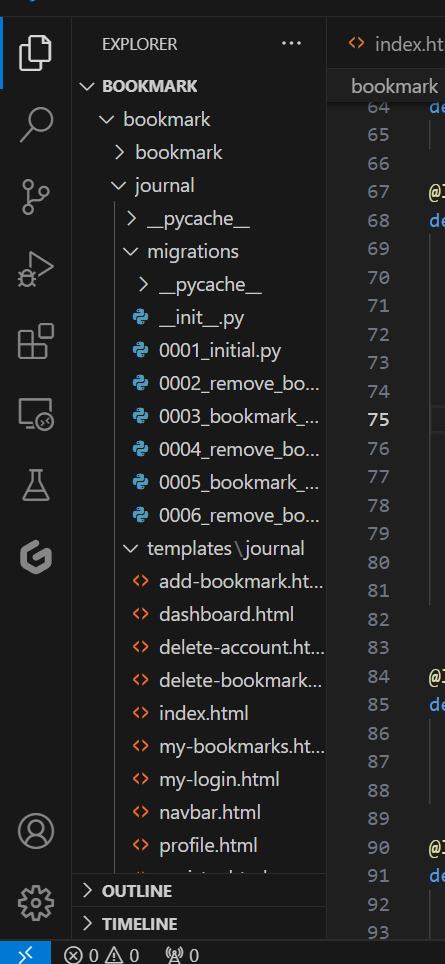


Fig. 6.1

****

**Fig 6.2**

**6.2.2 Implementing Login and Super-Admin Features** The next step was to develop the login page and super-admin functionalities:

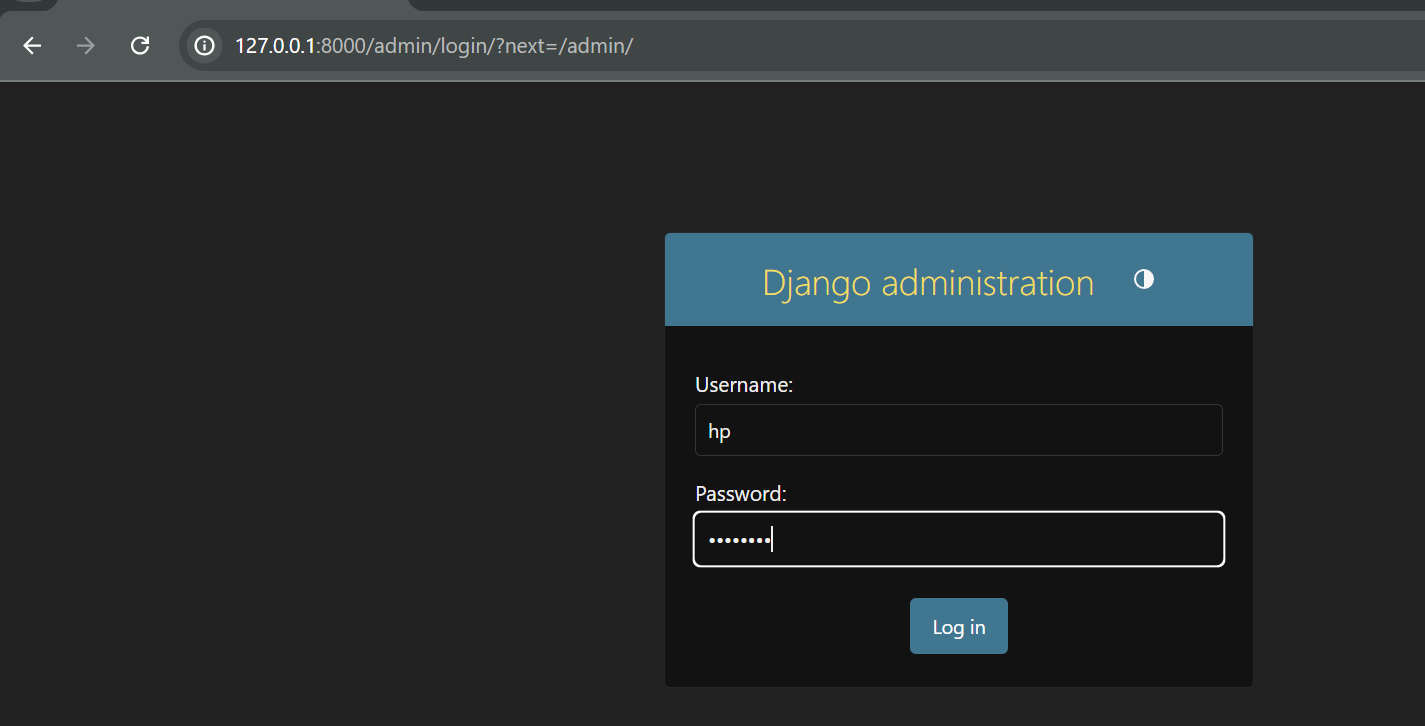


Fig 6.3 Here I have created a admin page and added User named HP.

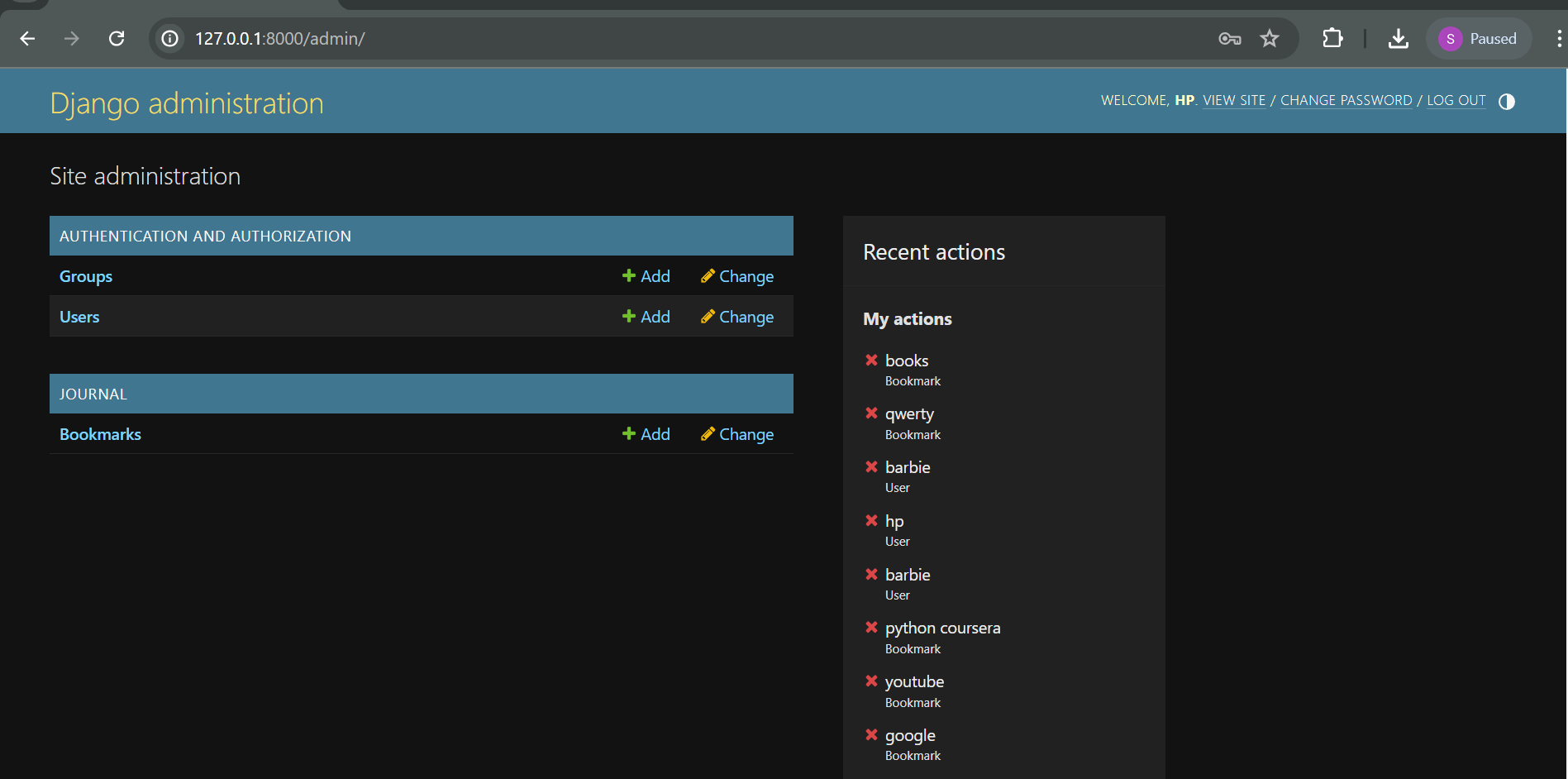
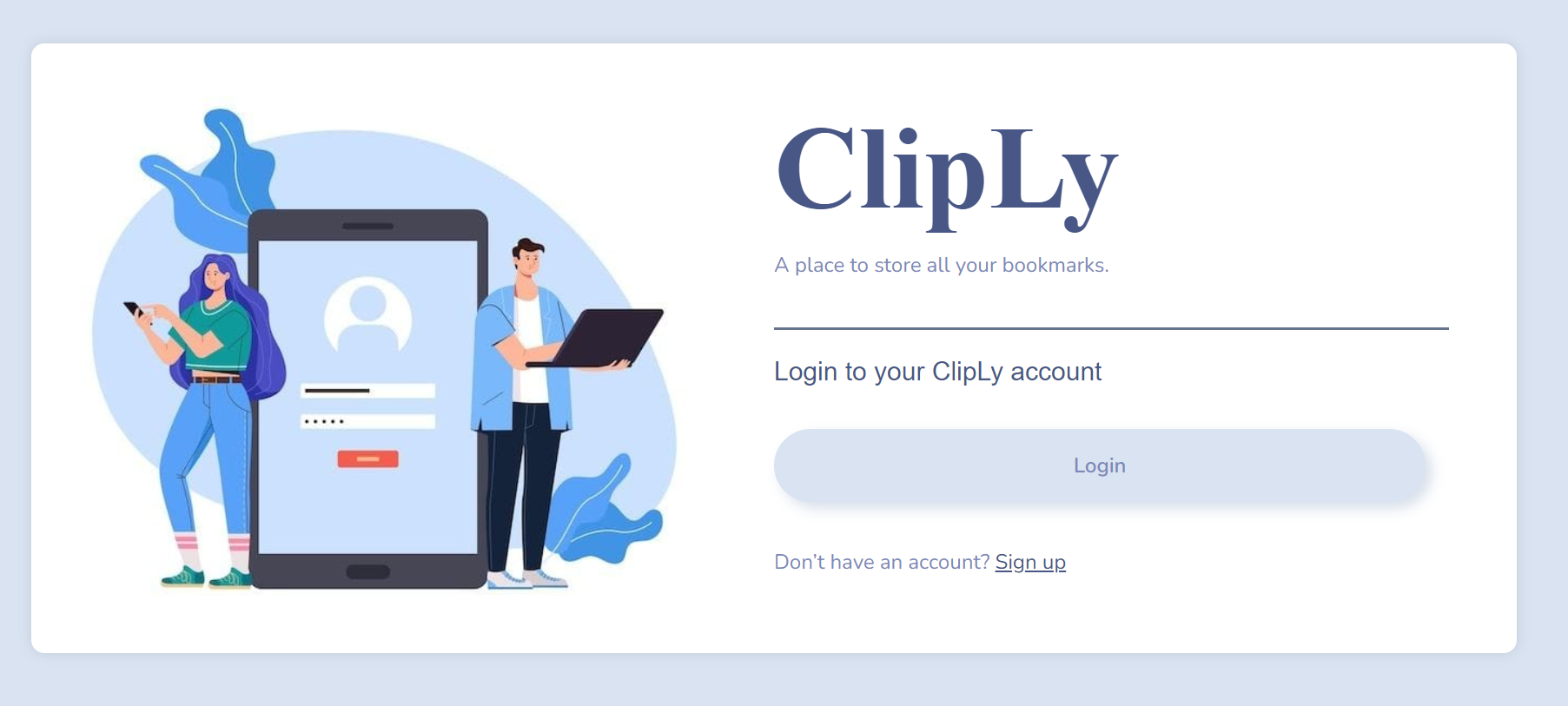


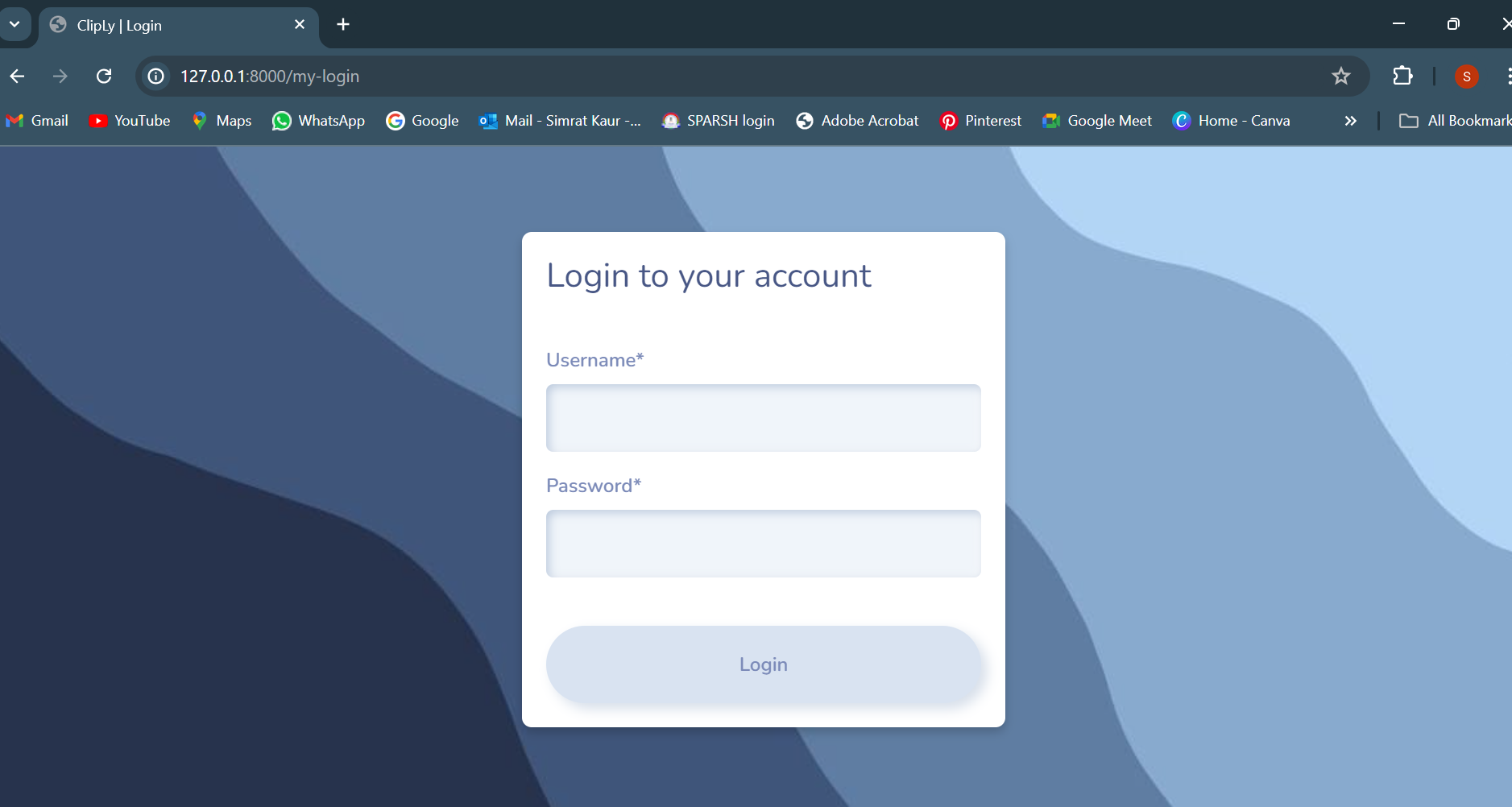
Fig 6.4 This figure shows the Django backend with user database and bookmark data-base

**6.2.3 Creating the Login Page**

* **Page URL:** /login
* **Components:**
  1. **Input Fields:** Two fields for email and password.
  2. **Login Button:** A button to submit the login form.
* **CSS Styling:** Applied a background design to enhance the visual appeal of the login page.



**Fig. 6.5**

****

**Fig 6.5**

**6.2.4 Implementing Sign-In Functionality**

* **Functionality:** Developed complete sign-in functions to authenticate users.
* **Redirection:** Users are redirected to the dashboard upon successful login.

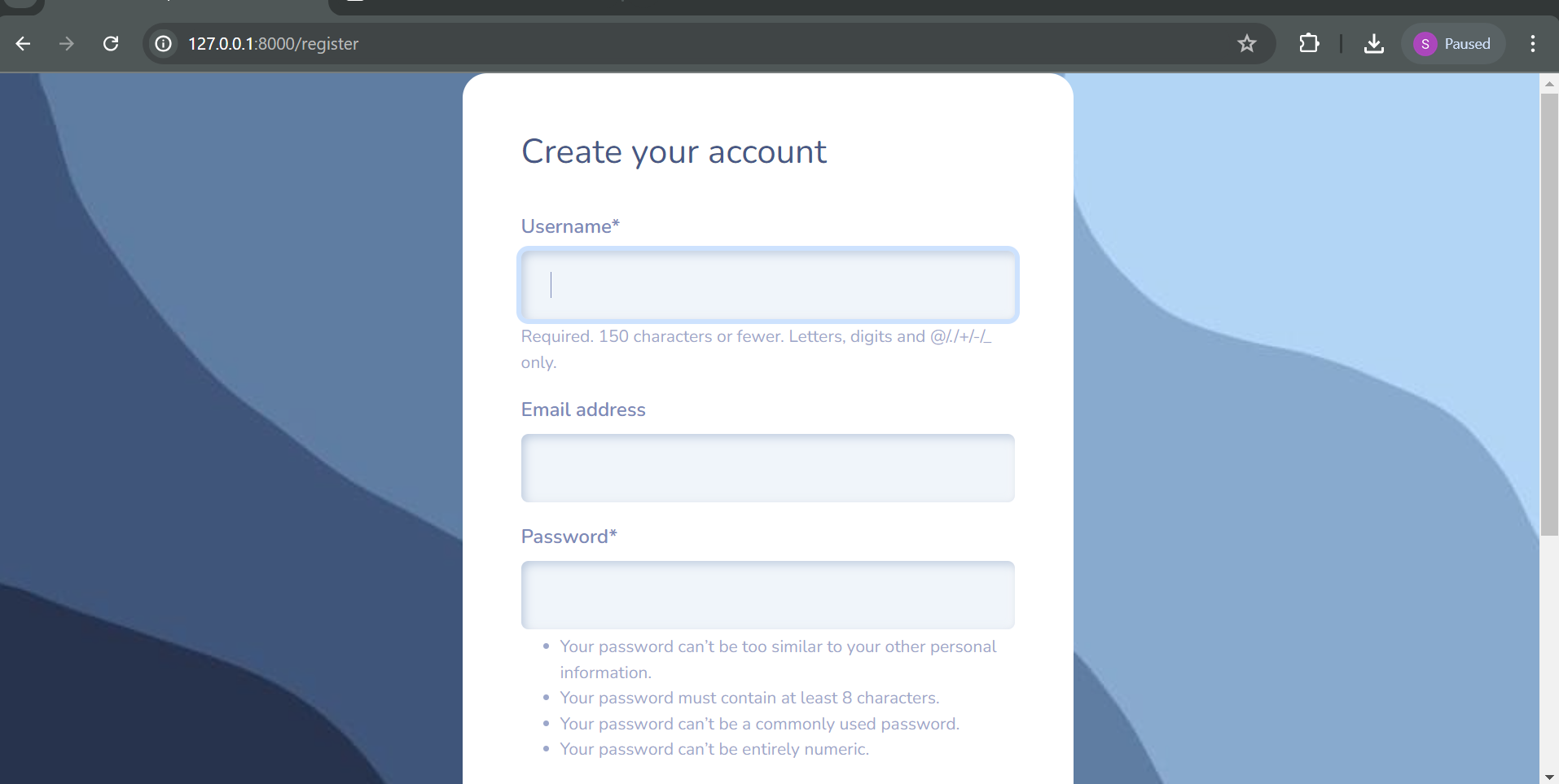


Fig 6.6

Here I have used pyhon django frameworks, built-in forms to create login forms.

**6.3 Dashboard Management**

**6.3.1 Creating the Dashboard Page**

* **Page Name:** home
* **Purpose:** Provides an overview of the website ClipLy.This page displays what is cliply, its objectives and why you should use it.

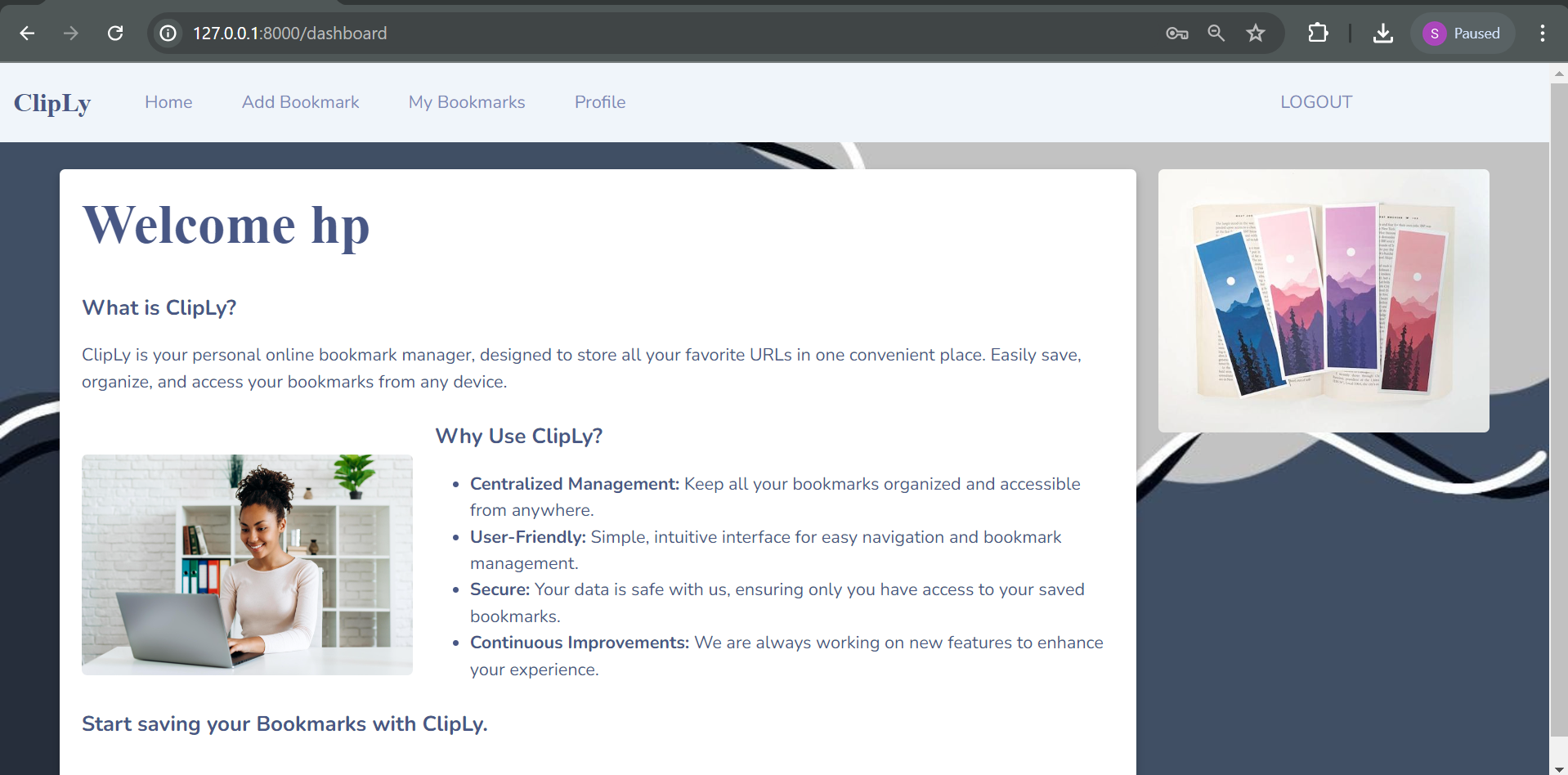
y

Fig 6.7

**6.3.2 Adding A Bookmark**

* **Create A Form :**
  1. **Form Page:** Designed a form for creating new bookmarks.
  2. **Database Integration:** Connected the form to a Django model to store tbookmark details in the database

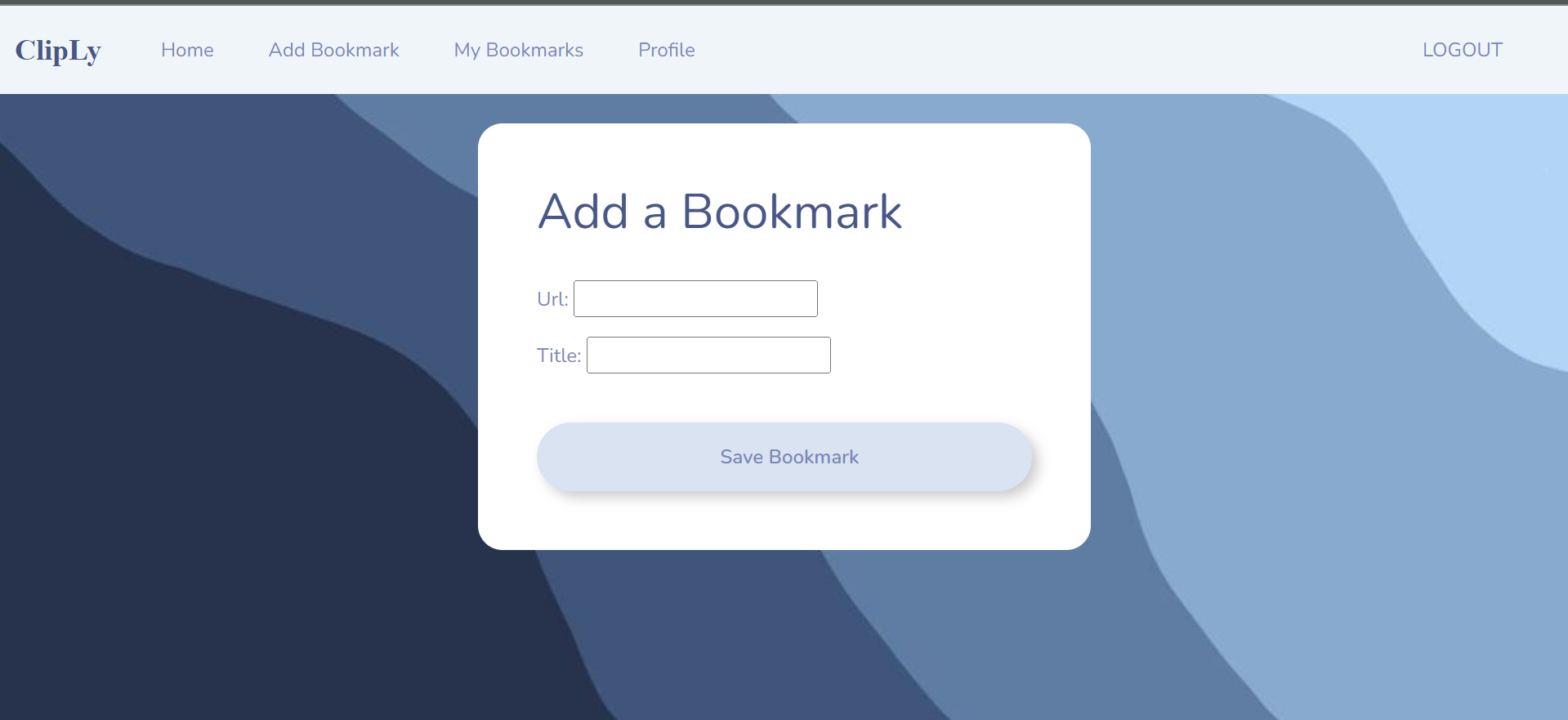


Fig 6.8

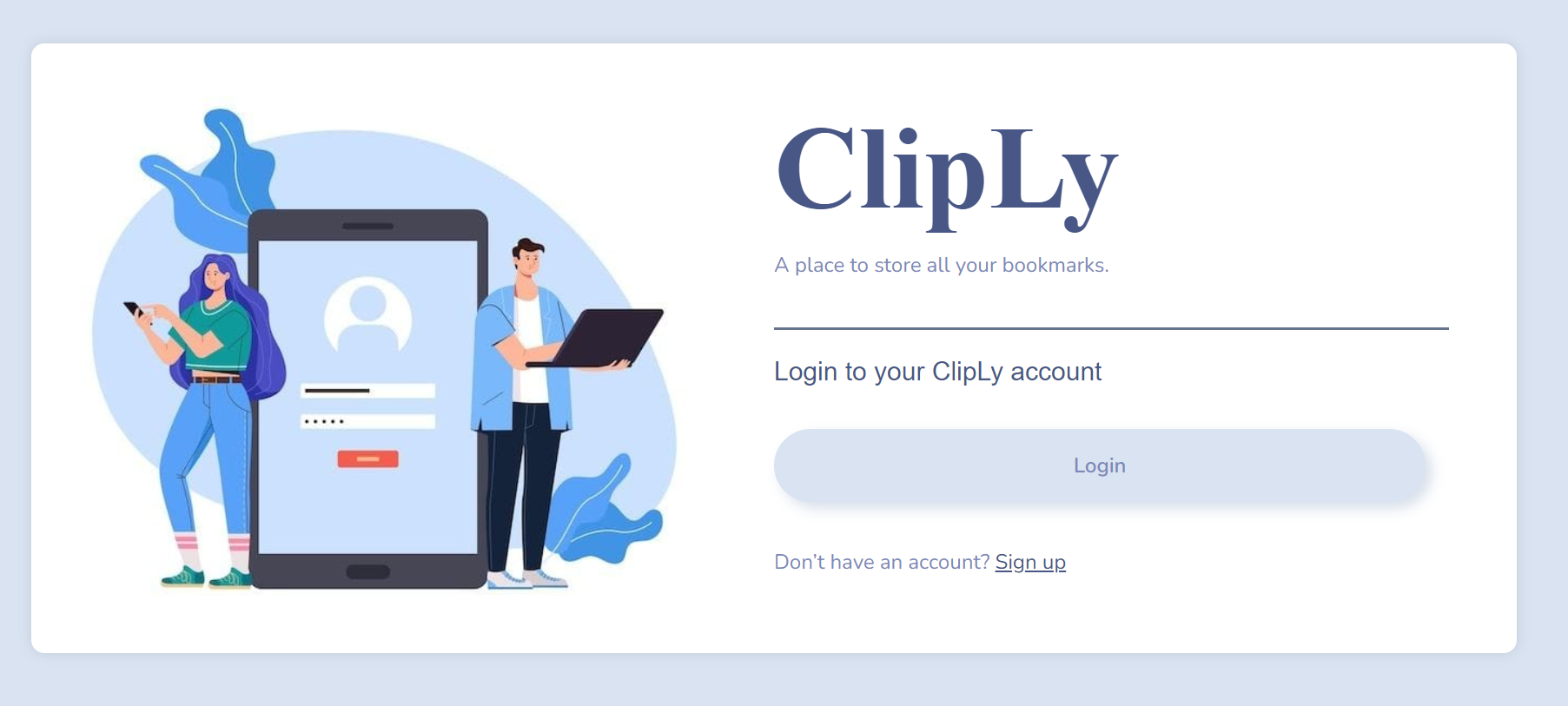
**6.3.3 Implementing Bookmark storage**

* **Functionality:** Stored bookmarks in Django SQL LITE database

**6.4 User Interface and Experience**

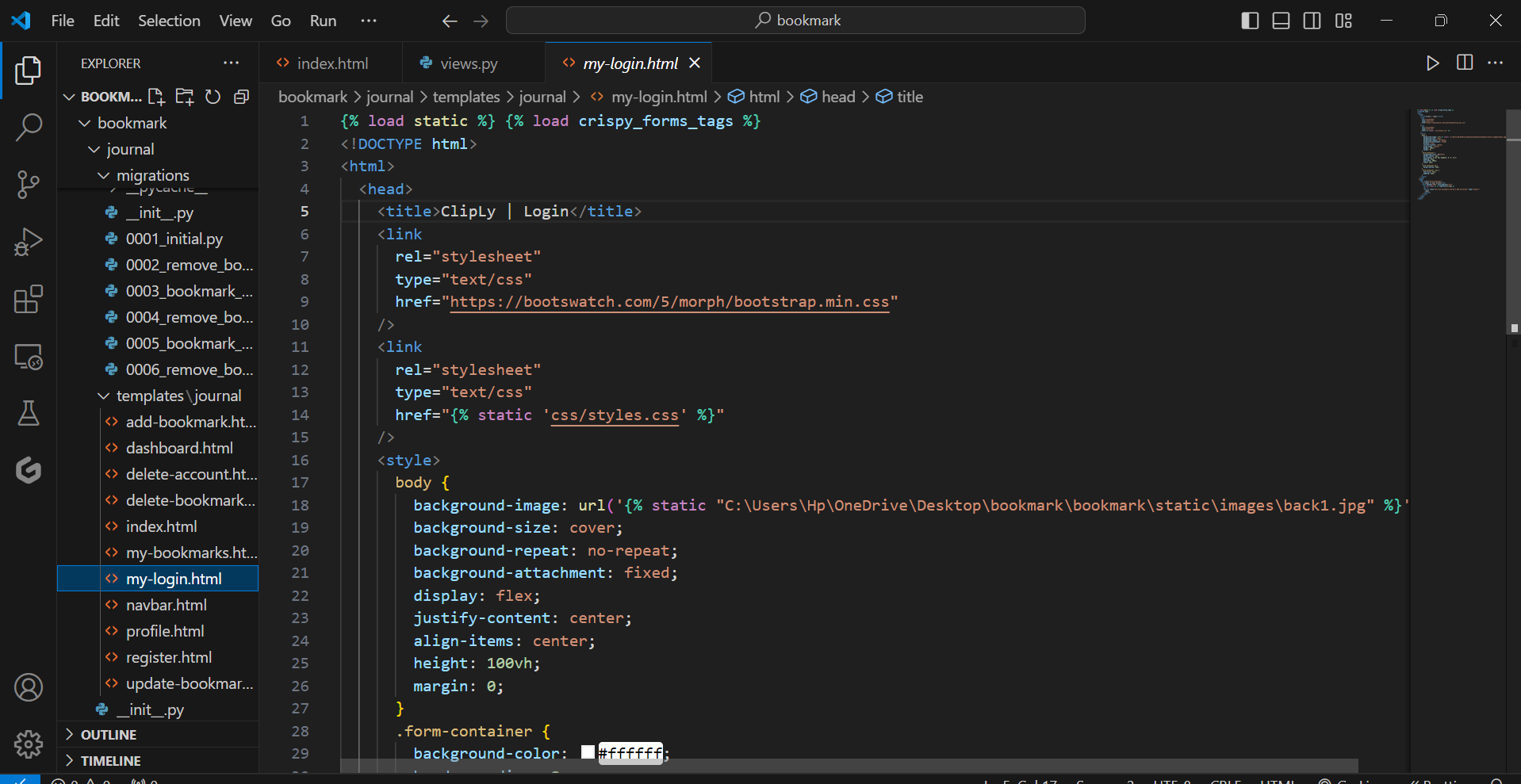
**6.4.1 Login Page Design**

* **Description:** The login page features a dark, textured background with a centered login box. The form includes fields for email and password, a "Signup" link, and a "Login" button.
* **Code and Output:**



**Fig 6.9:** Login Page Screenshot

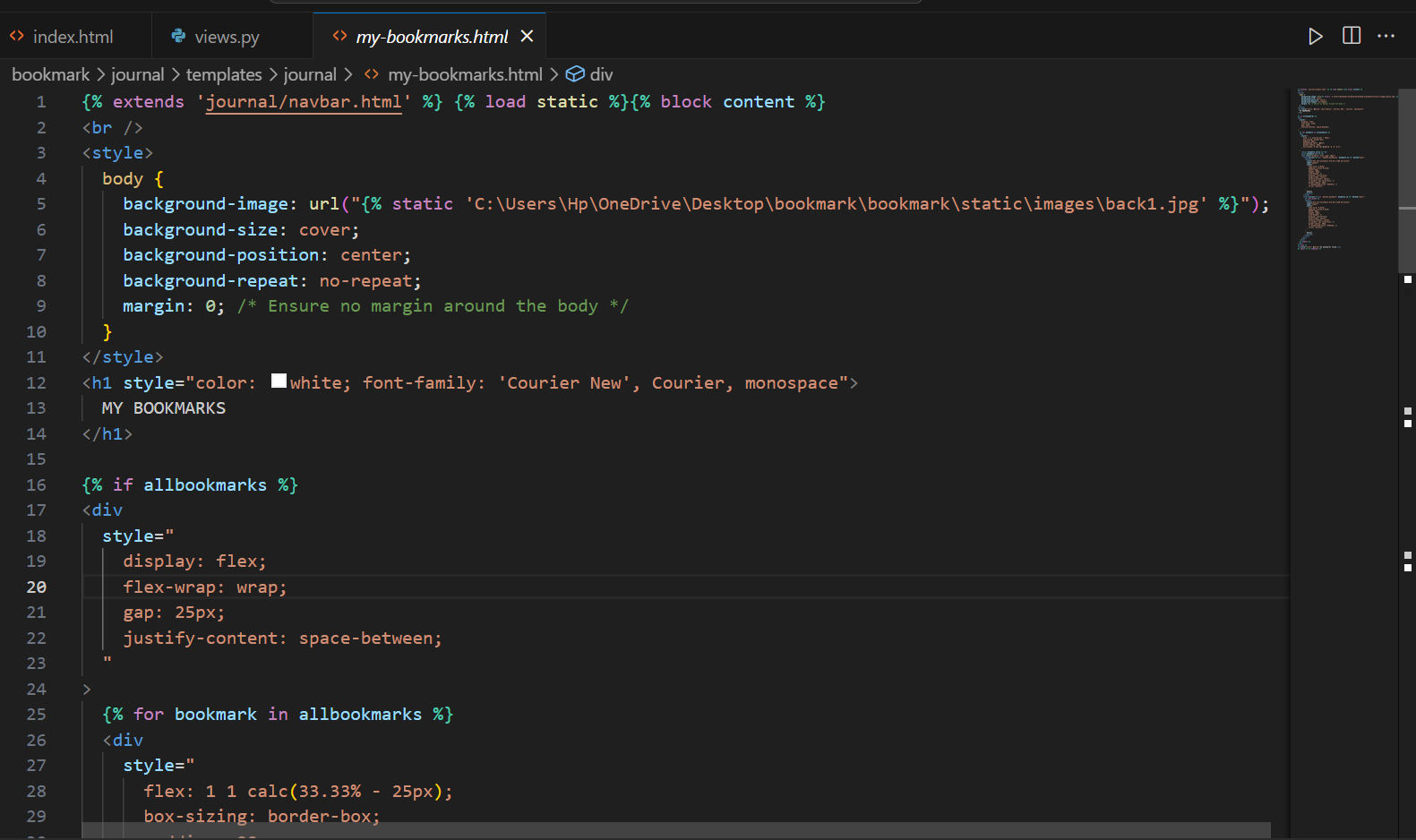
This code shows the HTML code of the my-login page in which I have used a flexbox with image insertion class and buttons that redirect you to specific page link.



**Fig 6.10:** Login Page Code

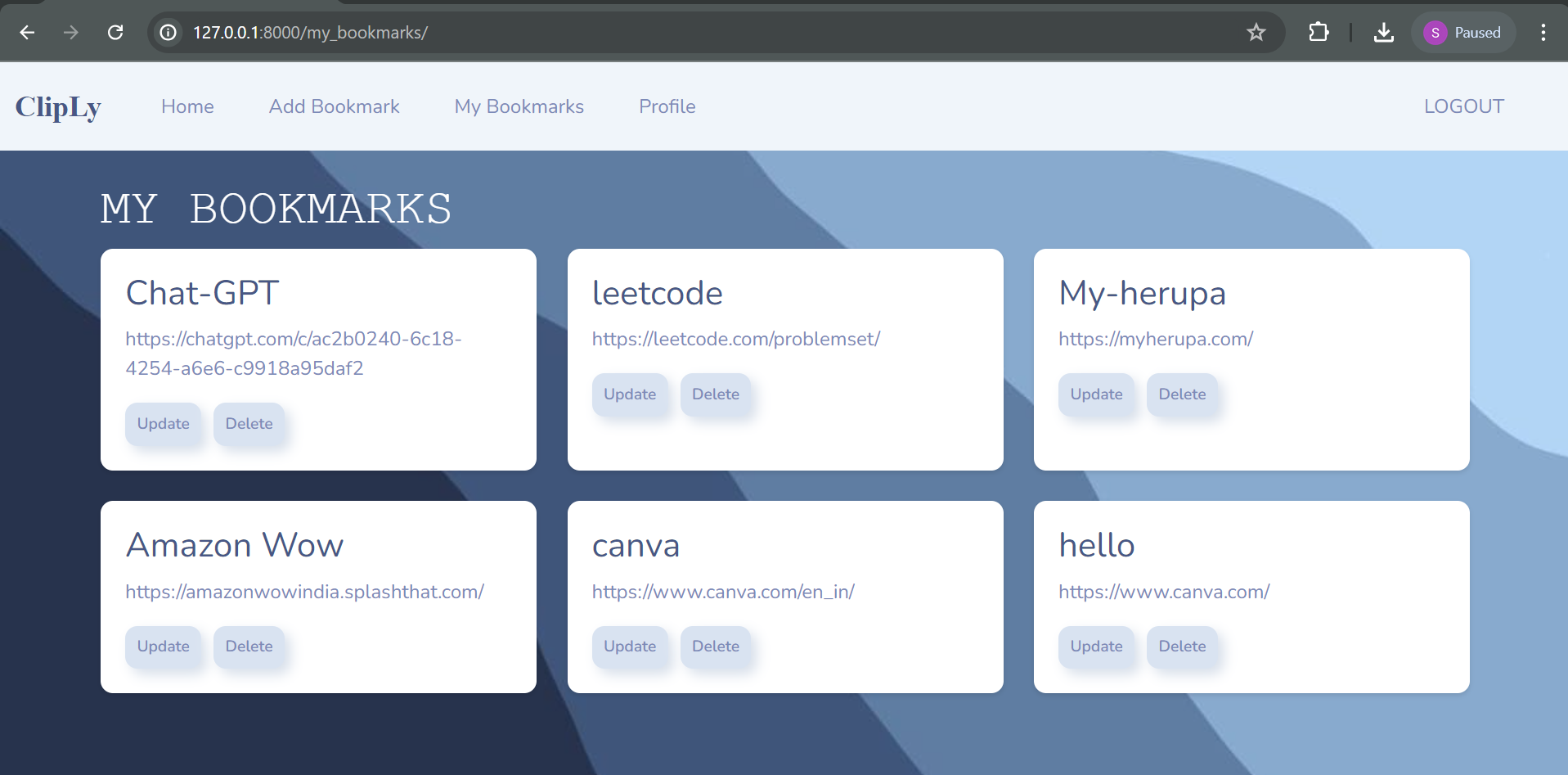
**6.4.2 Contact My bookmarks Page**

* **Description:** Displays a list of bookmarks saved in bookmarks
* **Code and Output:**



**Fig 6.11:** Contact Request Front-End Code

* The provided code is a Django template for displaying bookmarks in a visually appealing and user-friendly way. It extends a base template, navbar.html, and loads static files for styling. The main content block starts by setting a background image and styling the body to ensure it covers the entire screen without repeating. The .bookmark-container class is styled to organize bookmarks in a flexible, wrapping layout with space between items. Each bookmark is encapsulated in a .bookmark-card, styled with padding, a white background, rounded corners, and a shadow effect for a card-like appearance.
* The title of each bookmark is displayed using the .bookmark-title class, styled with a bold, monospace font. The .bookmark-buttons class positions buttons at the top right of each card, allowing users to edit or delete bookmarks. These buttons are further styled with the .btn-custom class for a consistent look and feel.
* Within the template, the allbookmarks context variable is checked. If it contains bookmarks, they are iterated over, and each bookmark's title and URL are displayed within a styled card. Edit and delete buttons are provided for each bookmark, linking to the appropriate URLs for updating or deleting the bookmark
  + **Fig 6.12: UI**



**6.5 User and Admin Management**

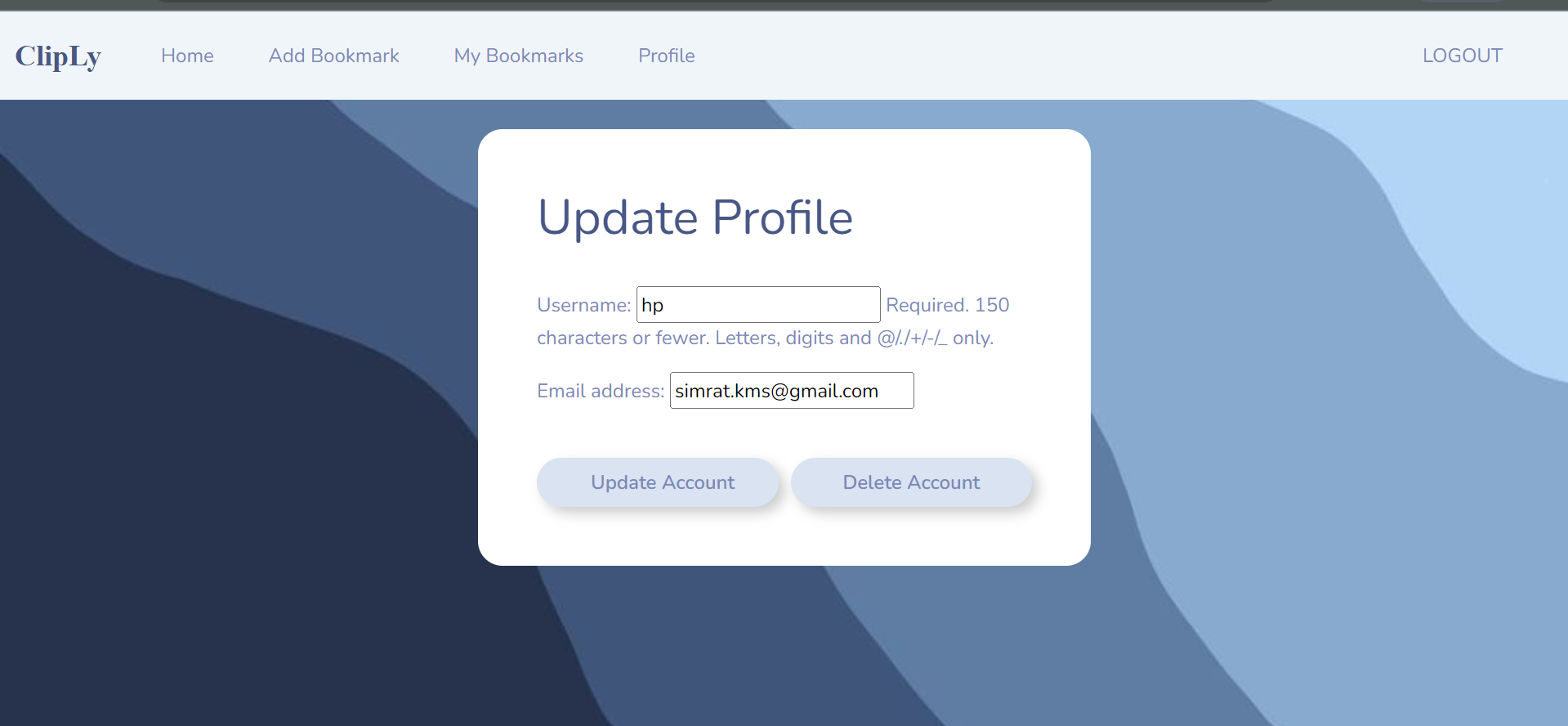
**6.5.1 Profile Page**

* **Description:** Displays user profile information and allows for editing personal details.
* **Code and Output:**
  + **Fig 6.13:** Profile Page



**6.5.2 User Management Interface**

* **Description:** Allows admins to add and manage user accounts with roles and permissions.
* **Code and Output:**
  + **Fig 6.14:** User Interface for updating user info



* + **Fig 6.15:** User Code

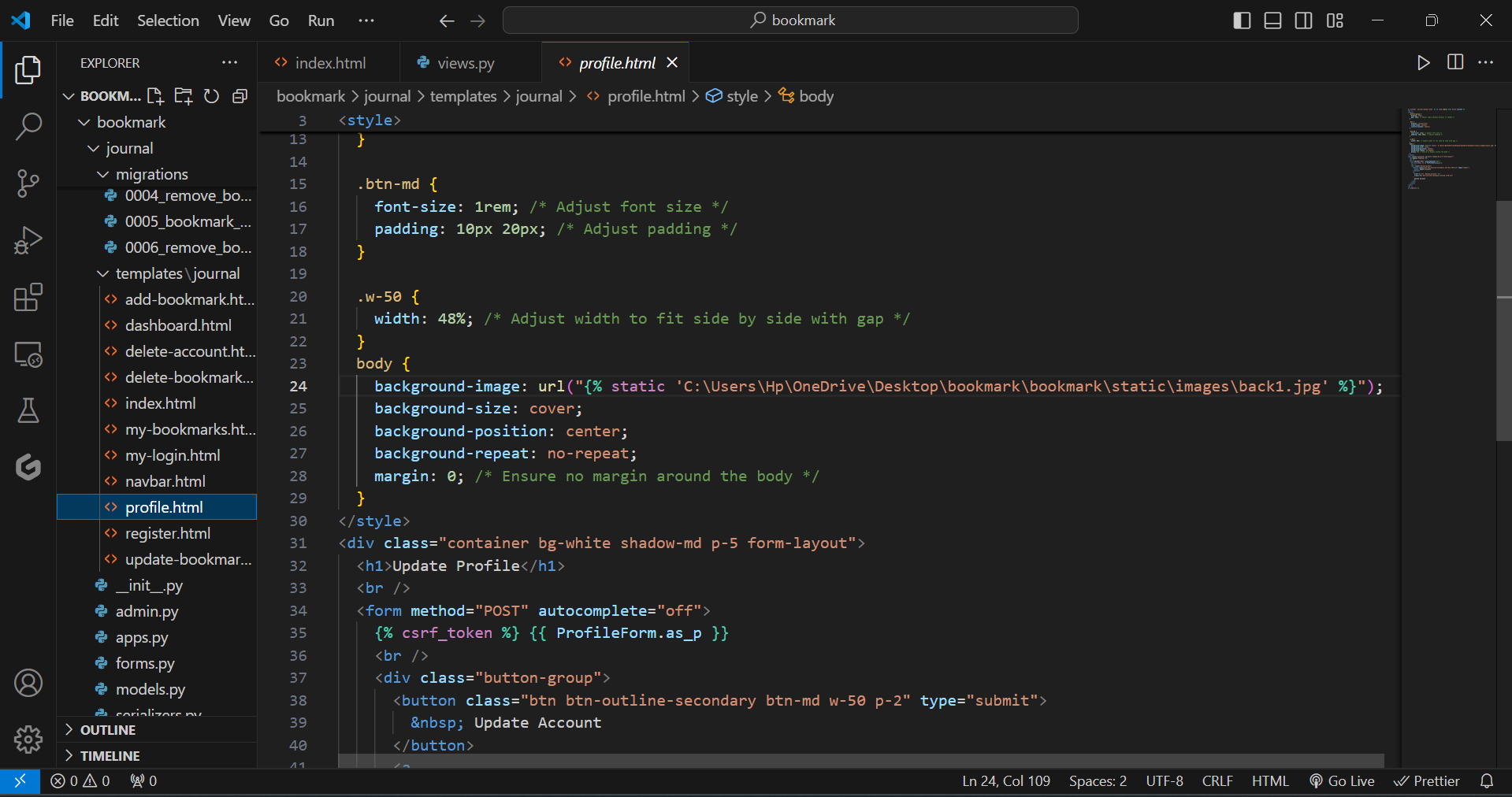


Fig 6.16

The provided code is a Django template for an "Update Profile" page. It extends the base template navbar.html and loads static files. The main content block starts with CSS styling to define a button group with flex layout and specific button styles for alignment and size. The body is styled with a background image that covers the entire page without repeating. A container <div> with classes for background color, shadow, and padding wraps the form elements

6.6 Modal Integration with chrome extension

**6.4 Technical Implementation**

**6.4.1 Extension Code**

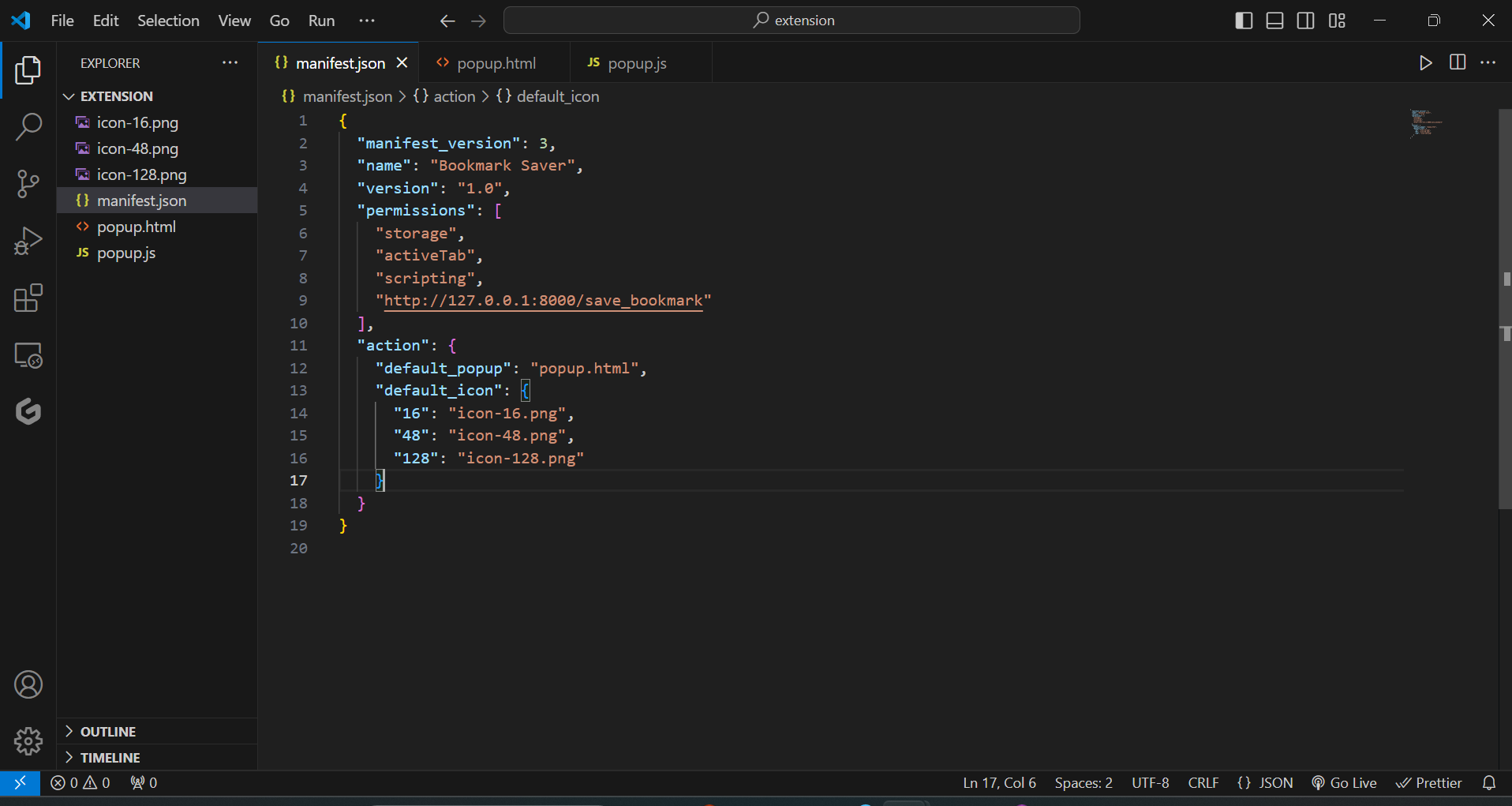


Fig 6.17

**6.4.1.1 Content Script**

* **Functionality:** Integrates with the web page to provide user interface components for the extension.

The provided code is an HTML document for a simple "Save Bookmark" interface. The `<style>` section defines the page's appearance, setting a dark background, rounded corners, and a shadow for the body. The heading is centered with white text, and the text input field is styled with padding, a border, and centered text. The button is styled with a background color, white text, and rounded corners, changing color on hover. The HTML body includes a heading, a text input for the bookmark title, and a button to save the bookmark, with a JavaScript file `popup.js` included for functionality.

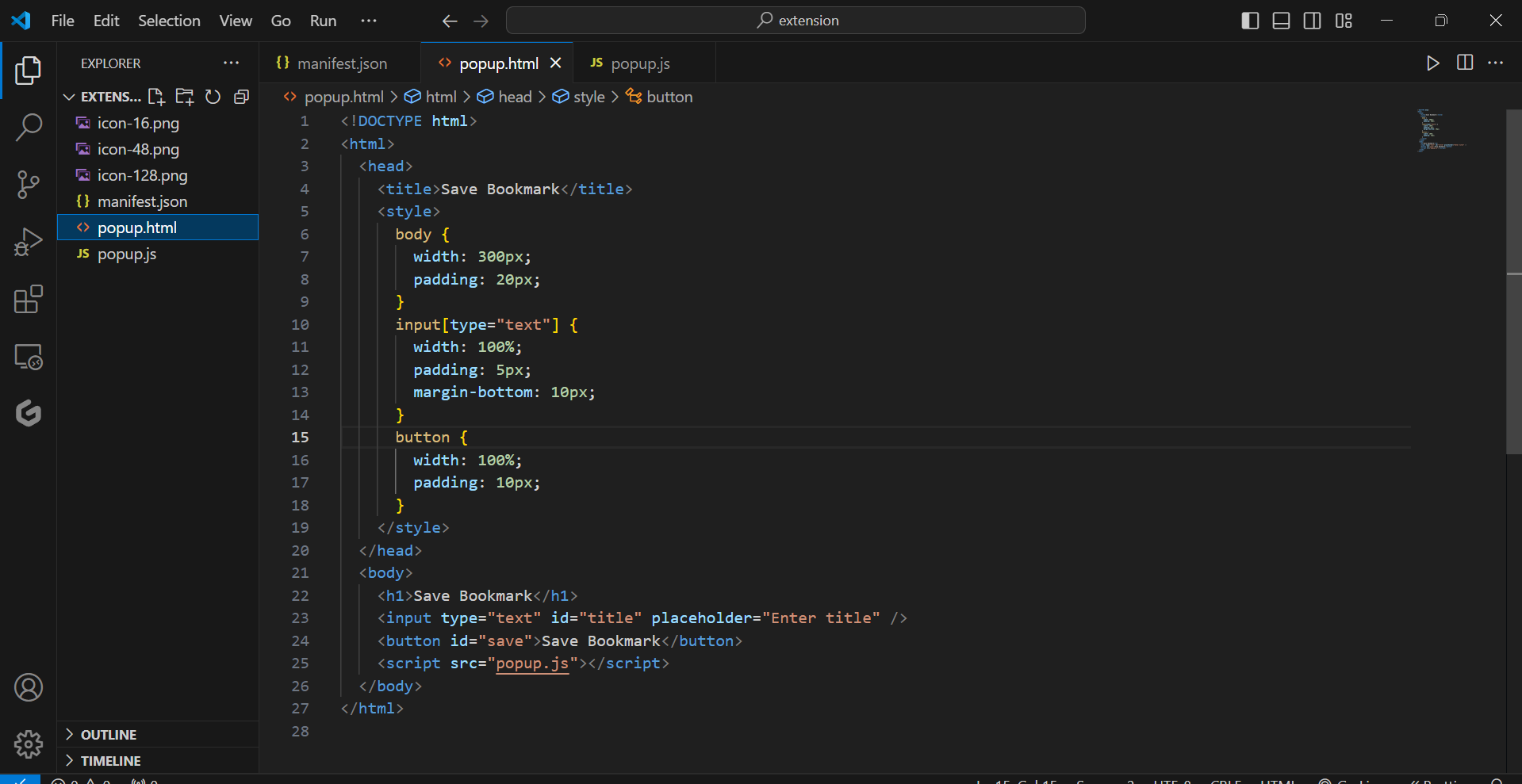
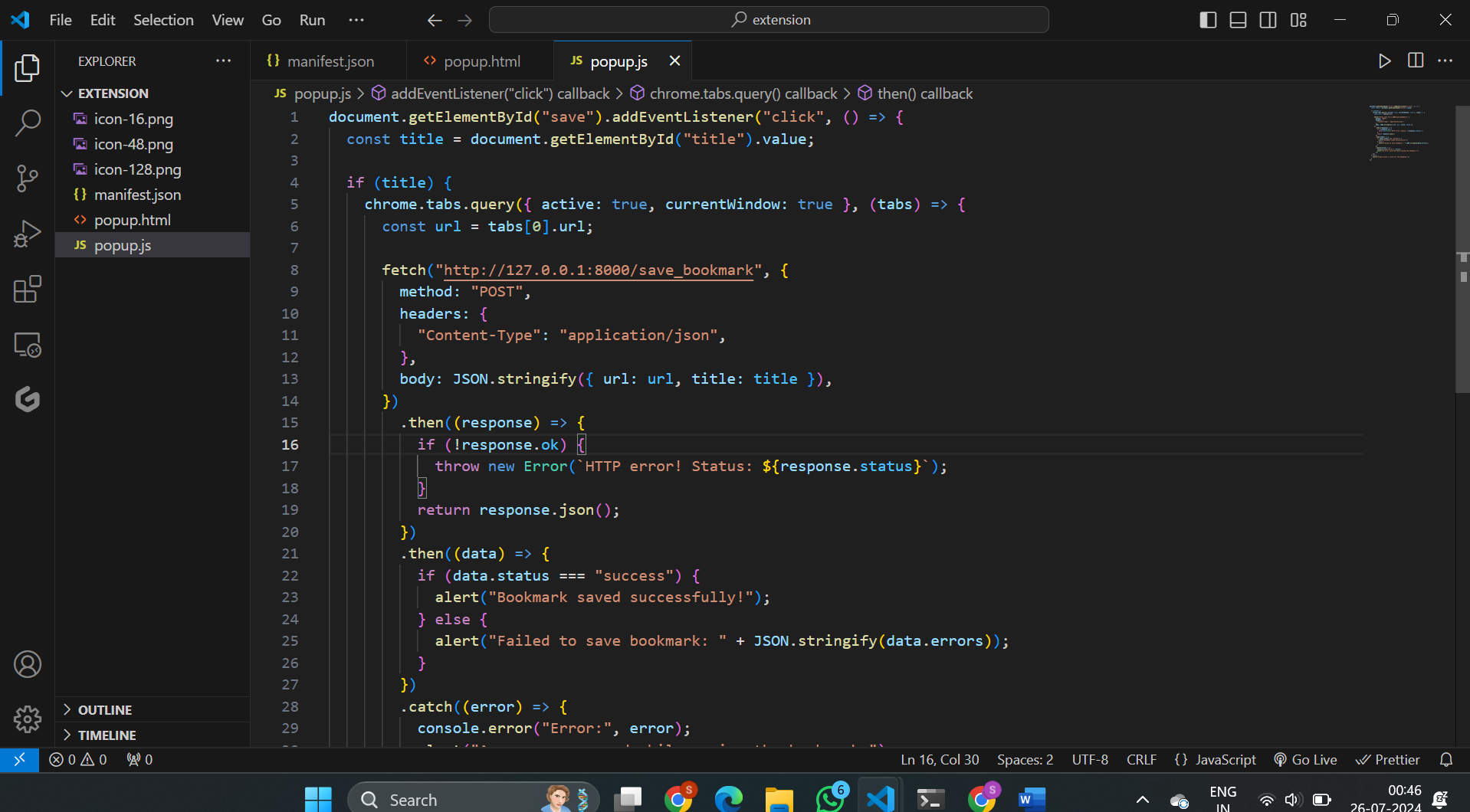


Fig 6.18

This JavaScript code adds a click event listener to the "Save" button. When clicked, it retrieves the value of the title input field and, if not empty, queries the active browser tab for its URL. It then sends a POST request to a local server endpoint to save the bookmark, including the URL and title in the request body. The response is processed to check if the bookmark was saved successfully, displaying an alert message accordingly. Errors during the process are caught and logged, and an alert is shown if there is an issue.



**Fig 6.19**

**Key Features:** Displays bookmark management options, interacts with background scripts.

**6.4.1.3 Popup Interface**

* **Functionality:** Provides a user-friendly interface for interacting with the extension's functionalities.

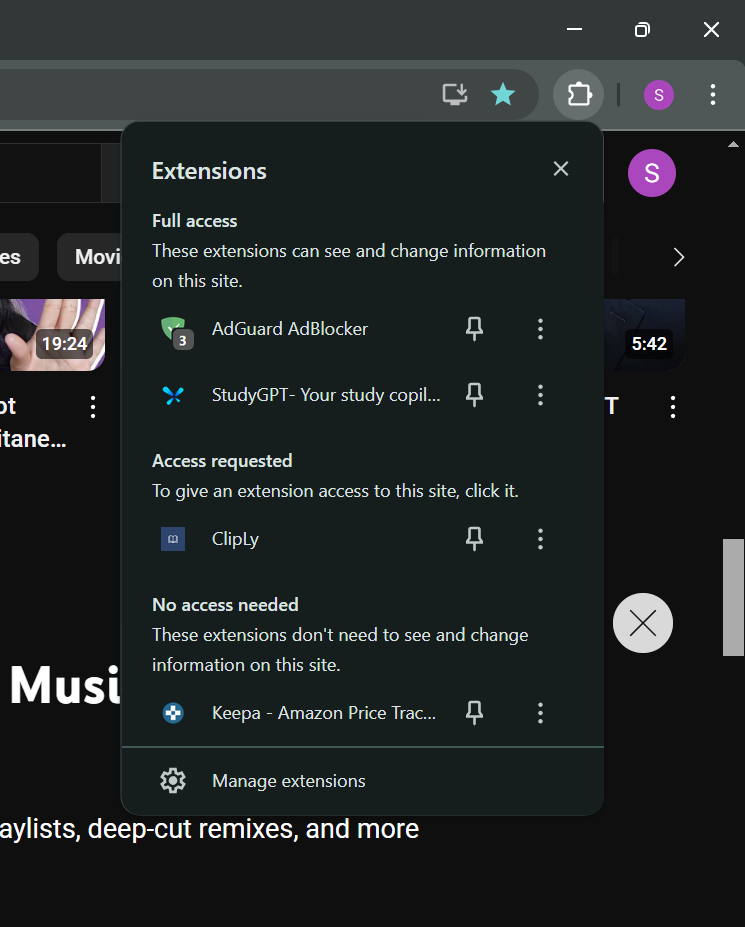


Fig 6.20

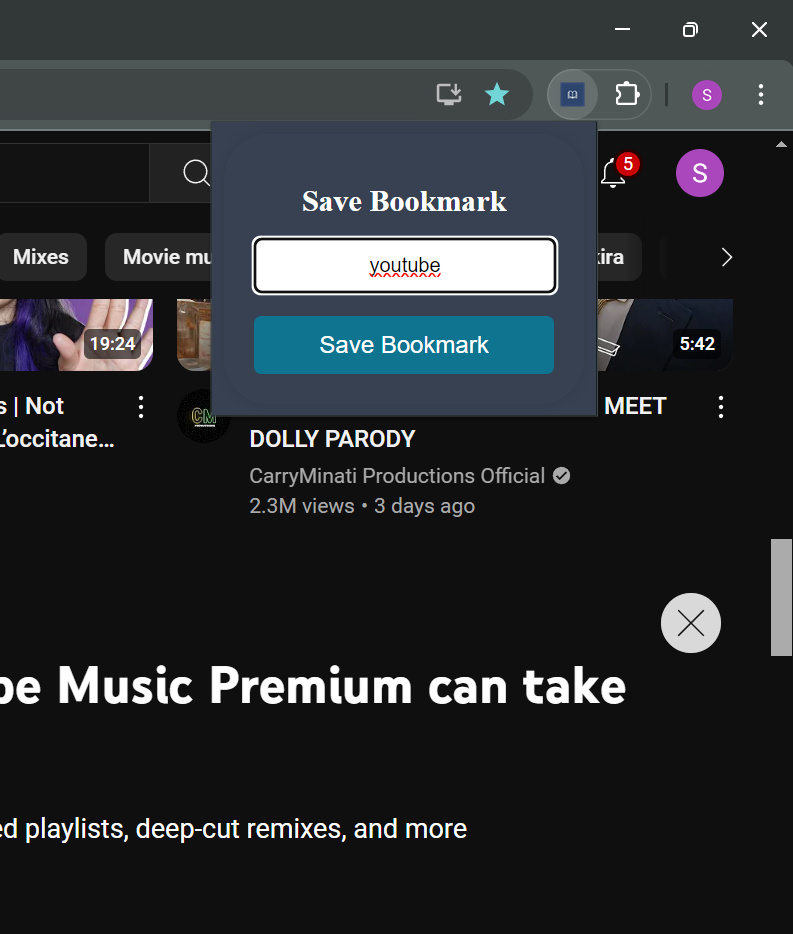
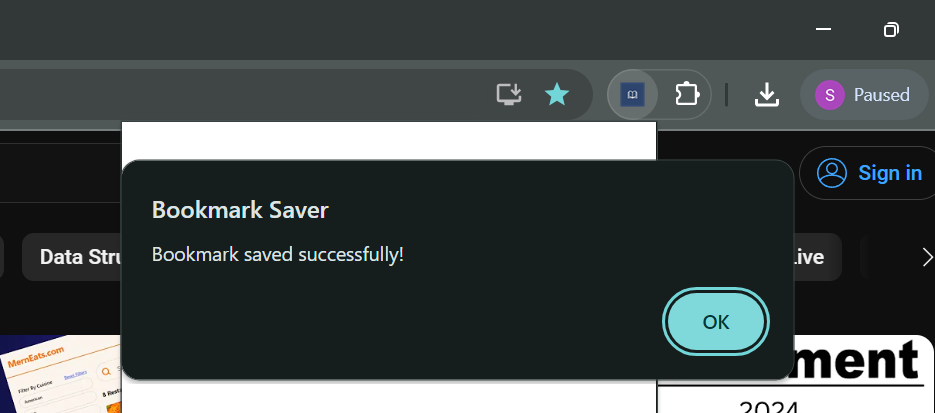


Fig 6.21



**Fig 6.22** Allows users to save their bookmarks.

**6.4.2.2 Models and Views**

* **Models:** Defined Django models for tickets, bookmarks, and user profiles.

This Django code defines a `Bookmark` model for storing bookmarks. It imports `models` from Django and `User` from `jango.contrib.auth.models`, although `User` is not used here. The `Bookmark` model includes two fields: `url`, a URL field for storing the bookmark’s URL, and `title`, a character field for the bookmark’s title. The `\_\_str\_\_` method returns the title of the bookmark, which helps in representing the object as a string.

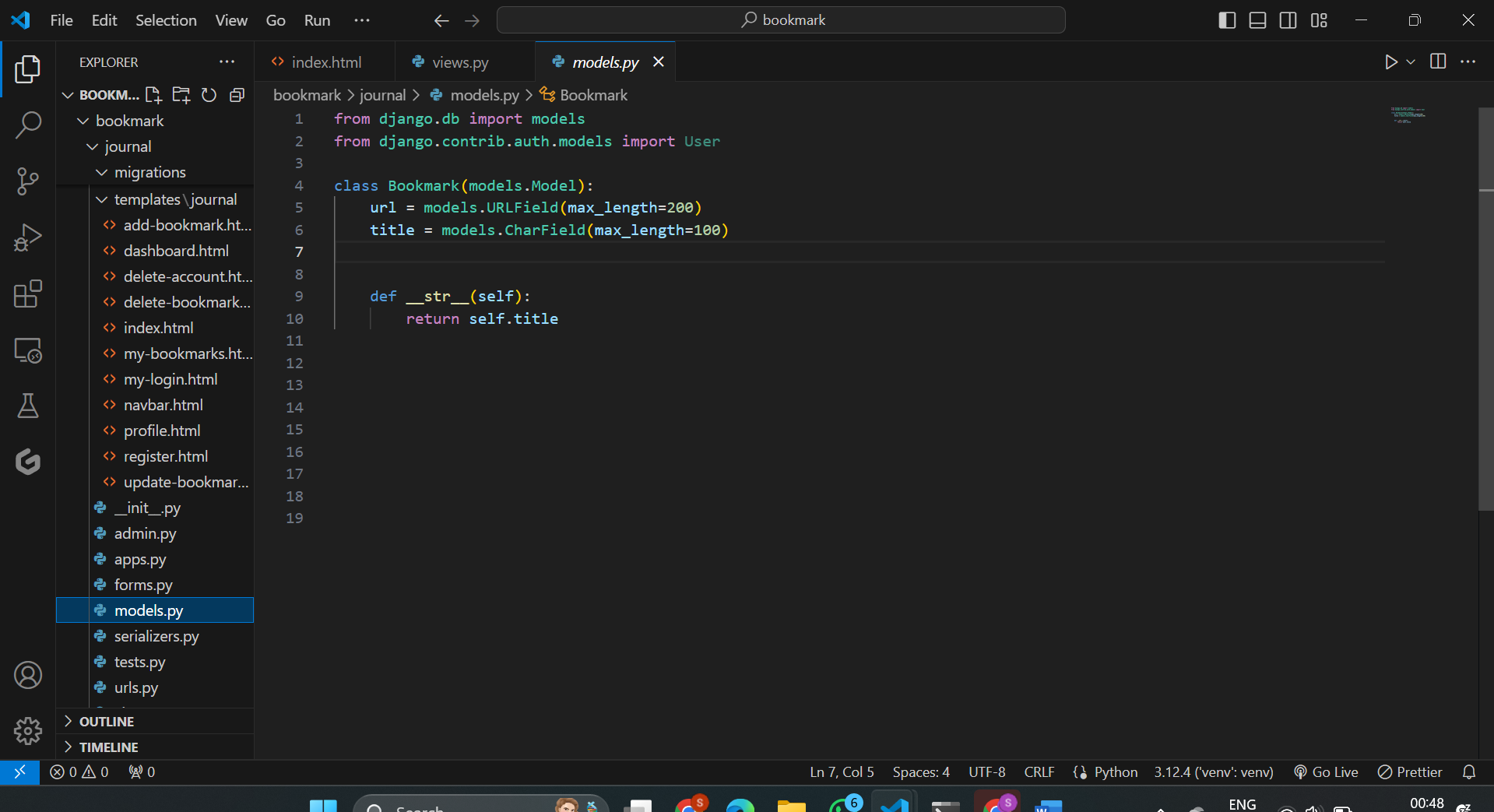


Fig 6.23

* **Views:** Created Django views to handle API requests and interact with the database.

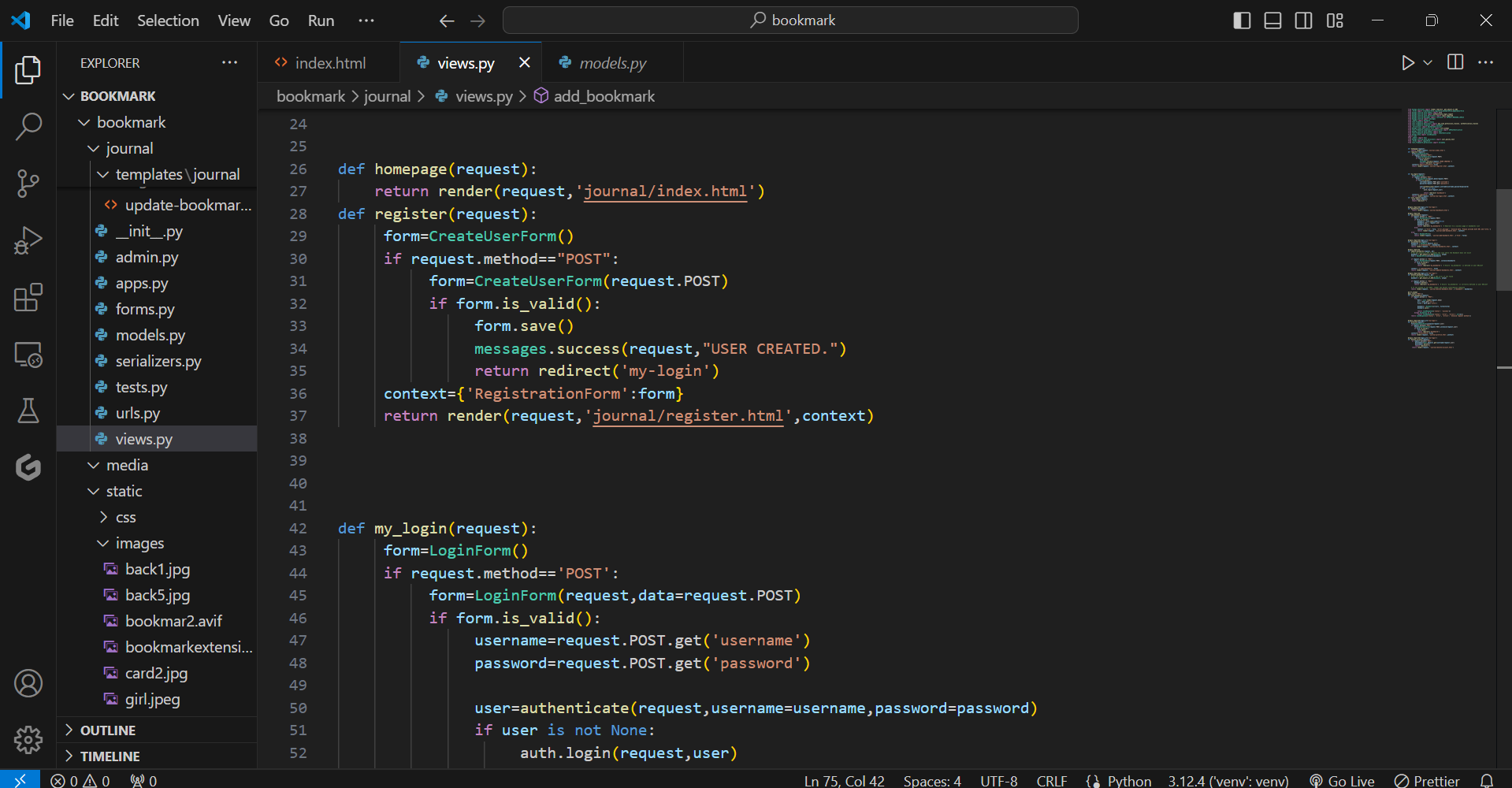


Fig 6.24

This Django code defines four view functions:

1. \*\*`homepage(request)`\*\*: Renders the `index.html` template for the homepage, handling GET requests.

2. \*\*`register(request)`\*\*: Manages user registration. It displays a registration form and, upon receiving a POST request with form data, validates and saves the new user. If successful, it shows a success message and redirects to the login page. If not, it redisplays the form.

3. \*\*`my\_login(request)`\*\*: Handles user login. It shows a login form and processes POST requests. On form submission, it validates the credentials, authenticates the user, logs them in if credentials are correct, and redirects to the dashboard. If authentication fails, it re-renders the login form.

4. \*\*`user\_logout(request)`\*\*: Logs out the current user and redirects to the homepage.

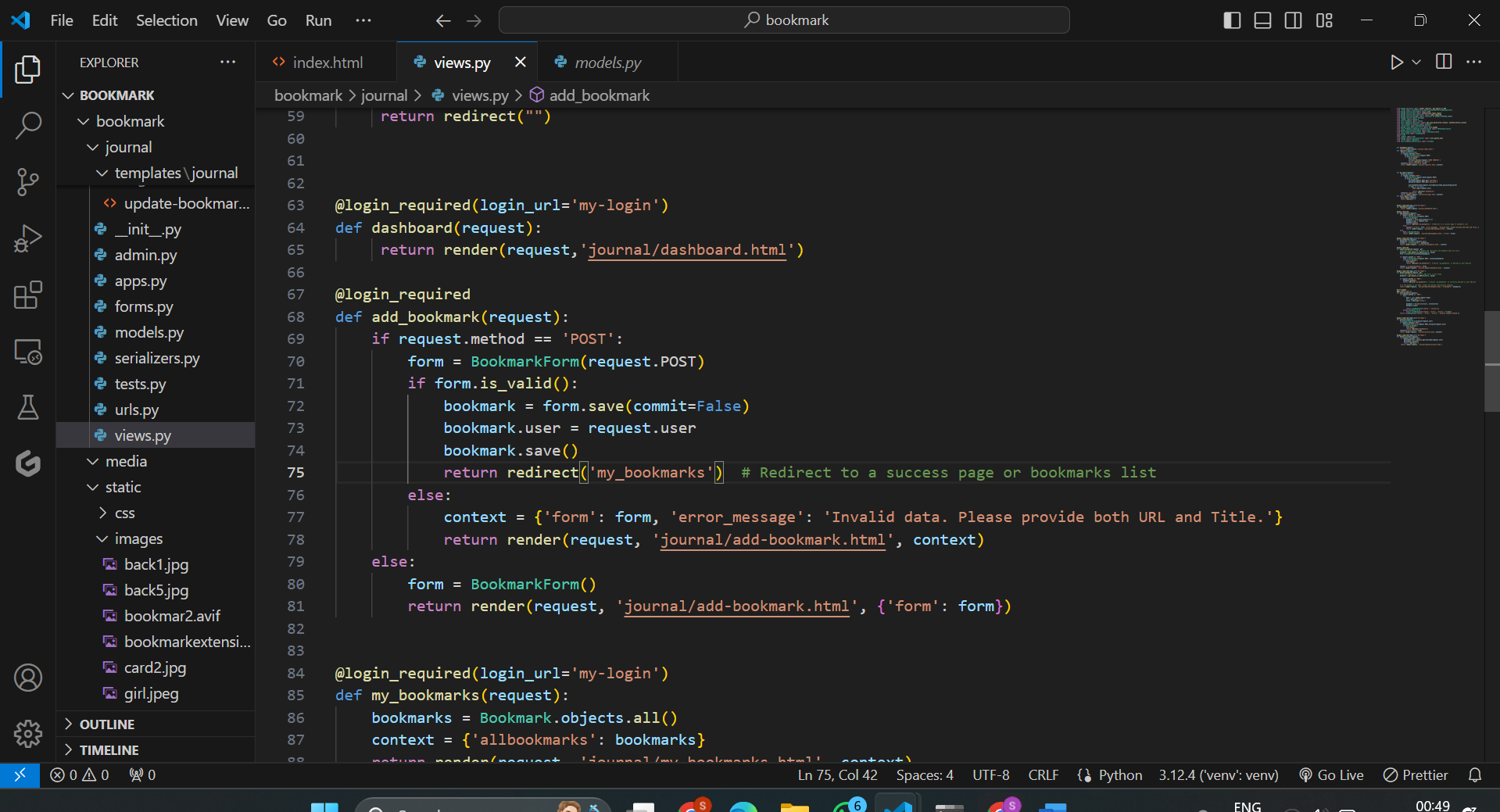


Fig 6.25

This code defines a Django view function add\_bookmark that allows authenticated users to add a new bookmark.

* **Decorator**: @login\_required ensures that only logged-in users can access this view.
* **POST Request Handling**: When the form is submitted via POST, it creates an instance of BookmarkForm with the submitted data. If the form is valid, it saves the new bookmark, assigning the current user to it, and then redirects to a page showing the user's bookmarks.
* **Invalid Form Handling**: If the form data is invalid, it re-renders the add-bookmark.html template with the form and an error message.
* **GET Request Handling**: For GET requests, it initializes a blank form and renders the add-bookmark.html template with this form.

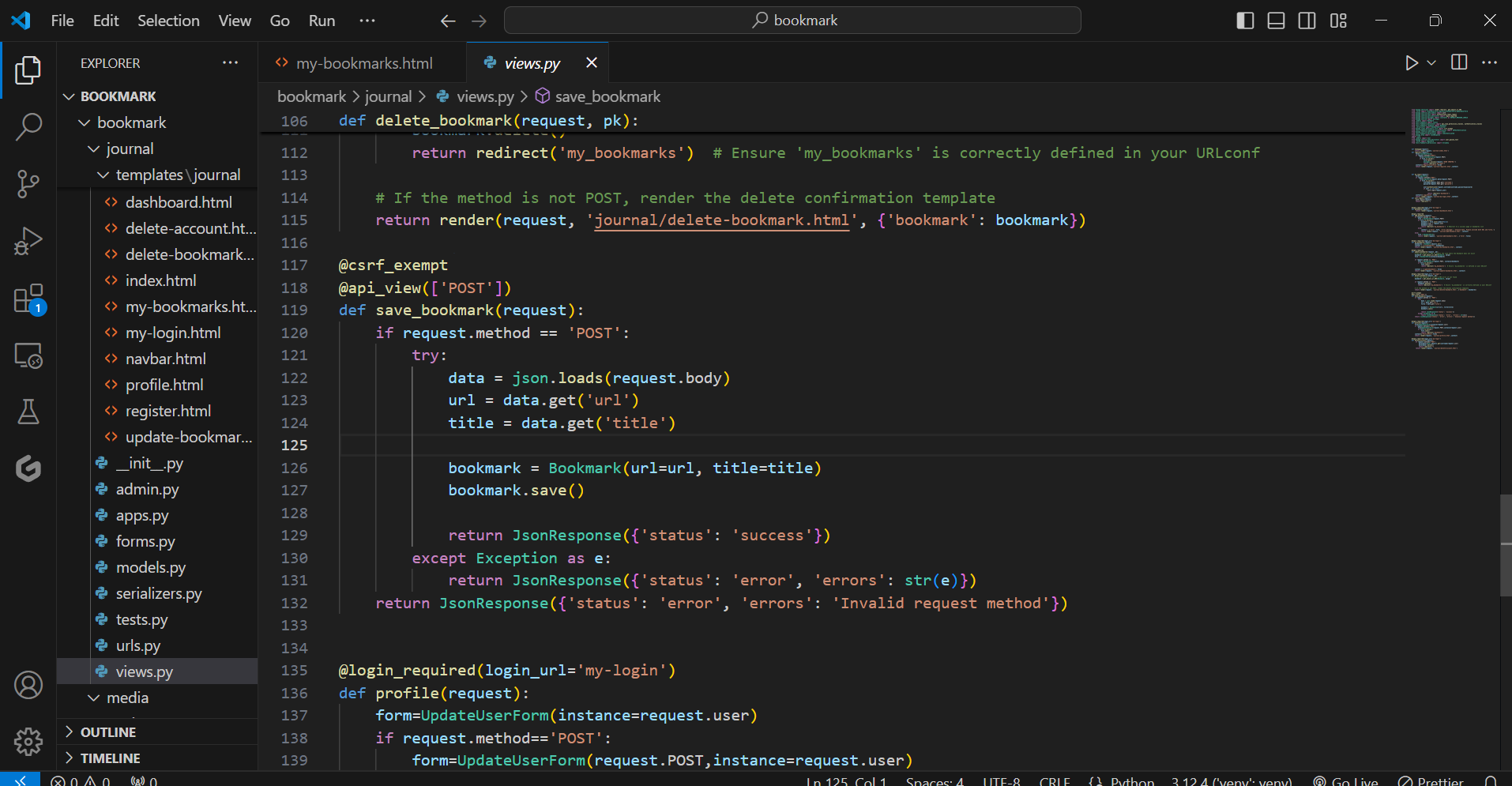


Fig 6.26

# This code defines a Django REST framework view function save\_bookmark for handling POST requests to save a bookmark.

# Decorators:

# @csrf\_exempt disables CSRF protection for this view, making it accessible from non-browser clients.

# @api\_view(['POST']) specifies that this view only accepts POST requests.

# POST Request Handling:

# It parses the JSON body of the request to extract url and title.

# Creates and saves a Bookmark object with these values.

# Returns a JSON response indicating success if the save operation is successful.

# Error Handling:

# Catches any exceptions during the process, returning a JSON response with an error status and a description of the issue.

# Invalid Request Method:

# Returns an error JSON response if the request method is not POST.

# 

# CHAPTER -7

**Chapter 7: Conclusion**

**7.1 Conclusion**

During my recent project, "Cliply," a bookmark saver Chrome extension developed using Django, I gained substantial experience in integrating web technologies and browser extensions. This project allowed me to apply my theoretical knowledge of web development and Django in a practical context, significantly enhancing my technical skills.

The Cliply extension provides users with an intuitive interface to save, organize, and manage their bookmarks directly from their browser. Leveraging Django for the back end ensured robust functionality for user authentication, bookmark management, and data storage. The extension's design focuses on ease of use, providing features for quick bookmark saving and effective organization.

Working on this project involved various stages, including requirement analysis, design, development, testing, and deployment. This experience not only improved my coding capabilities but also my understanding of project management, teamwork, and effective communication in a professional setting. The feedback received from users and peers was invaluable in refining the project and delivering a high-quality product. This project marks a significant advancement in my career, offering practical insights into both browser extension development and Django web applications.

**7.2 Suggestions**

To further enhance the Cliply extension, consider implementing a feedback mechanism to regularly collect user opinions and suggestions. This will help in continuously improving the extension’s features and user experience. Conducting usability testing can identify and address any issues, ensuring that the extension remains intuitive and user-friendly.

Ensuring full responsiveness and compatibility across various browser versions and devices will enhance accessibility and user engagement. Performance optimization is crucial to minimize resource-heavy operations, resulting in quicker load times and smoother functionality.

Regular security audits should be conducted to identify and address potential vulnerabilities, ensuring the extension is secure and complies with data protection regulations.

**7.3 Future Scope**

Future enhancements for the Cliply extension could include:

1. **Mobile Application Development:** Develop a companion mobile application for Android and iOS platforms, allowing users to manage bookmarks on their smartphones seamlessly.
2. **Category based Tagging:** In the Bookmark Saver application, category-based tagging can be done to organize bookmarks by assigning them to specific categories or tags. This feature enhances the management and retrieval of bookmarks by grouping them into relevant categories, making it easier for users to find and sort their saved links. By implementing this functionality, users can categorize their bookmarks based on topics, interests, or any other criteria, improving the overall user experience and efficiency of the application.
3. **Multi-Language Support:** Add multi-language support to cater to a global audience, making the extension accessible to users from various linguistic backgrounds.
4. **Enhanced Security Features:** Introduce advanced security measures such as two-factor authentication and end-to-end encryption to ensure data security and user privacy.
5. **Scalability and Performance Optimization:** Optimize the extension and its back-end infrastructure to handle increased usage and ensure smooth performance during peak times.