

CHAPTER 1 - INTRODUCTION TO PROJECT

1.1 Project

Comic Corner is an online platform that offers a vast collection of comics, manga, bande dessinée, and light novels alike. Users can create account, login, and browse through a plethora of options available on the website. The platform provides comprehensive details about each comic, including its publisher, writer, description, and other relevant information. Users can also read the comics online without any hassle.

1.2 Purpose

The purpose of Comic Corner is to provide a comprehensive online platform for comic book enthusiasts, manga fans, and readers of light novels. By offering a vast collection of titles, detailed information about each comic, and an easy-to-use interface, Comic Corner aims to create a one-stop destination for anyone interested in exploring the world of comics.

Through its user-friendly features and immersive experience, Comic Corner encourages users to discover new titles and genres, and to engage with like-minded readers. Whether you are a long-time fan or a newcomer to the medium, Comic Corner offers a platform to access and enjoy a diverse range of comics, manga, bande dessinée, and light novels.

In summary, the purpose of Comic Corner is to foster a community of comic book lovers and to provide them with a comprehensive and engaging online platform to explore, read, and discover their favorite titles.

1.3 Objective

The objective of Comic Corner is to become the go-to online platform for comic book enthusiasts, manga fans, and readers of light novels. To achieve this objective, Comic Corner aims to:

1. Offer a vast and diverse collection of titles to cater to the interests of all users.
2. Provide detailed and accurate information about each comic, including its publisher, writer, description, and other relevant information.

3. Create an immersive and engaging experience for users, allowing them to read comics online and explore their favorite genres.
4. Foster a community of like-minded readers, encouraging them to interact, share their thoughts and discover new titles.
5. Continuously improve and update the platform based on user feedback and changing trends in the comic book industry.

By meeting these objectives, Comic Corner aims to establish itself as the leading online destination for comic book enthusiasts and to provide them with a comprehensive and engaging platform to access and enjoy their favorite titles.

1.4 Scope

The scope of the Comic Corner project includes the development of an online platform that provides a comprehensive collection of comics, manga, bande dessinée, and light novels. The website will have the following features:

1. User Registration and Login: Users can create an account and login to access the website's features and functionalities.
2. Search and Browse: Users can search for their favorite titles or browse through different genres and categories to discover new titles.
3. Detailed Information: The website will provide comprehensive details about each comic, including its publisher, writer, description, and other relevant information.
4. Reading Comics Online: Users can read comics online through the website's interface, without the need for any external applications or software.
5. Future User Interactions: In the future, users will be able to interact with the website by sharing their thoughts and ratings on the comics they have read. They can also add books to their favorite list, bookmark pages to continue where they left off, and access other accessibility features.
6. Admin Panel: An admin panel will be created to manage the website's content, users, and other settings.

The scope of the project includes the development of the website's front-end and back-end, database management, user authentication and authorization, and other necessary functionalities. The website will be designed to provide an immersive and engaging experience for users, with an easy-to-use interface and intuitive navigation. The project's scope also includes testing, debugging, and maintenance to ensure the website's smooth

operation and continued improvement. The future features of user interaction will be developed and implemented in subsequent phases of the project.

1.5 Technology and Literature Review

Technology Review:

The Comic Corner project is built using the MERN stack, which includes MongoDB, Express.js, React.js, and Node.js. Here's a brief overview of each technology:

1. **MongoDB:** MongoDB is a NoSQL document-oriented database that stores data in JSON-like documents. It is a popular choice for web development because of its scalability, flexibility, and high performance.
2. **Express.js:** Express.js is a popular web application framework for Node.js that simplifies the development of web applications. It provides a set of features and tools for building web applications, including routing, middleware, and templates.
3. **React.js:** React.js is a JavaScript library for building user interfaces. It allows developers to create reusable UI components and build complex UIs using a declarative approach. It is widely used for building single-page applications and dynamic user interfaces.
4. **Node.js:** Node.js is a JavaScript runtime built on the V8 engine. It allows developers to run JavaScript code on the server-side, making it possible to build fast, scalable, and high-performance web applications.

By using the MERN stack, the Comic Corner project is able to provide a fast, scalable, and responsive web application that is easy to maintain and update.

Literature Review:

The use of technology in the comic industry has evolved significantly over the years. With the rise of digital media, comic book publishers have started to embrace new ways of delivering content to readers. Here are some key trends in the industry:

1. **Digital Comics:** The rise of digital comics has made it easier for readers to access and read comics from anywhere, at any time. Many publishers now offer digital versions of their comics, which can be accessed through various platforms and devices.

2. **Webcomics:** Webcomics have become increasingly popular in recent years. They are usually published online and can be read for free or for a small fee.

Webcomics are often self-published and allow creators to reach a wider audience.

3. **Augmented Reality:** Some comic book publishers have started to experiment with augmented reality technology to enhance the reading experience. By using AR, readers can interact with the comic in new and exciting ways, such as by scanning images with a smartphone camera to reveal hidden content.

4. **Social Media:** Social media has become an essential part of the comic industry. Creators and publishers use social media platforms to promote their work, engage with fans, and build communities around their comics.

By incorporating some of these trends and technologies, the Comic Corner project can provide a modern and engaging platform for readers to discover and enjoy comics.

1.6 Project Planning

1.6.1 Project Development Approach and Justification

The Comic Corner project was developed using an agile development approach. The project was divided into several sprints, each of which focused on a specific set of features and functionalities. The agile approach allowed for flexibility and adaptability throughout the development process, as the project requirements and priorities evolved over time.

- During each sprint, the development process followed the following steps:
- **Planning:** The project requirements and priorities were reviewed, and a plan was developed for the sprint.
- **Development:** I worked on implementing the planned features and functionalities.
- **Testing:** The features and functionalities were tested to ensure that they met the project requirements and were functioning correctly.
- **Review:** The completed sprint was reviewed, and feedback was gathered from stakeholders and users.
- **Refinement:** The feedback was used to refine the project requirements and priorities for the next sprint.

The agile development approach allowed for continuous improvement and adaptation throughout the development process, ensuring that the project met the evolving needs of stakeholders and users.

1.6.2 Project Effort, Time and Cost Estimation

Effort Estimation:

$$\text{Effort} = \text{Number of weeks} * \text{Number of hours per week}$$

$$\text{Effort} = 12 * 48$$

$$\text{Effort} = 576 \text{ hours (Approx)}$$

Time Estimation:

$$\text{Time} = \text{Effort} / \text{Number of hours per day}$$

$$\text{Time} = 576 / 8$$

$$\text{Time} = 72 \text{ days (Approx)}$$

Cost Estimation:

$$\text{Cost} = \text{Effort} * \text{Hourly rate}$$

$$\text{Cost} = 576 * 5 \text{ (hosting and domain charge average cost)}$$

$$\text{Cost} = \$2,880$$

The estimation is based in several assumptions, and may vary as those factors change. It's important to keep in mind that these are only estimations and that actual values may differ.

1.6.3 Roles and Responsibilities

I act as a full stack developer in this project, handling the front-end and the back-end and synchronizing the data flow within parts of the project.

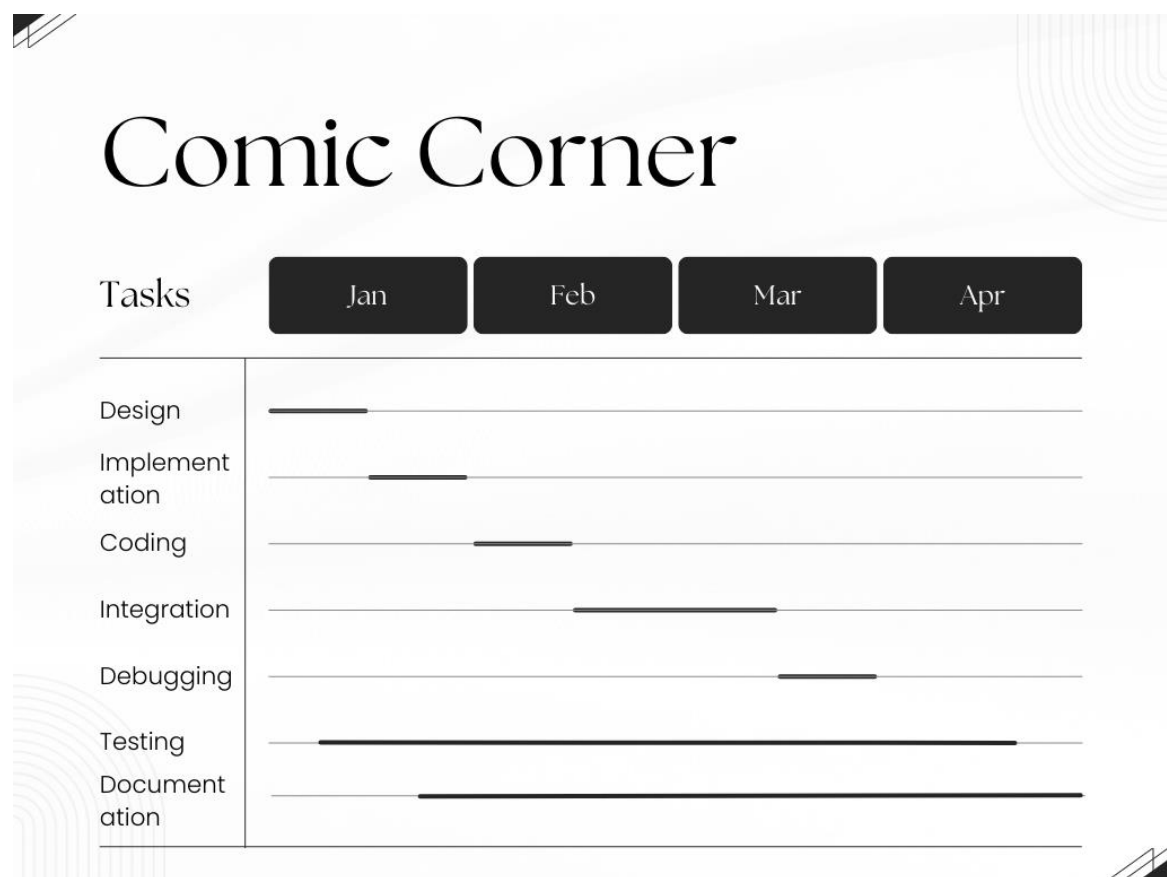
1.6.4 Group Dependencies

1. The development of the user interface: The tasks involved in designing and developing the user interface, such as wireframing, design, and implementation, are dependent on each other. If the wireframes are not completed, the design cannot be created, and if the design is not ready, the implementation cannot begin.
2. The development of the database: The tasks involved in designing and developing the database, such as schema design, table creation, and data population, are dependent on each other. If the schema design is not completed, the table creation

cannot begin, and if the table creation is not complete, the data population cannot begin.

3. The development of the comic reading feature: The tasks involved in developing the comic reading feature, such as page rendering, zooming, and bookmarking, are dependent on each other. If the page rendering is not completed, zooming cannot be implemented, and if bookmarking is not ready, users cannot save their progress.

1.7 Project Scheduling



2. System Analysis

2.1 Study of Current System

The study of the current system involves analyzing the existing process of browsing and reading comics online. In today's digital age, there are several websites and applications that offer users access to comics and manga online. Some of the popular websites include Crunchyroll, ComiXology, and Manga Rock.

The current systems offer a wide range of comics and manga from different genres and publishers. Users can browse through the available titles and read them online. These systems offer various features, such as bookmarking, saving favorite comics, and recommendations based on the user's preferences.

However, some of the current systems have limitations, such as a limited selection of titles or difficulty in finding specific titles. Some systems also have a cluttered user interface, making it difficult for users to navigate and find what they are looking for. Additionally, some systems require a subscription or payment to access certain titles or features.

The study of the current system provides valuable insights into what works and what needs improvement in online comic systems. This information can help in the development of a better system that addresses the shortcomings of the existing systems and provides a better user experience.

2.2 Problem and Weaknesses of Current System

While the current systems for browsing and reading comics online offer a wide range of titles and features, there are also several problems and weaknesses that need to be addressed. Some of the main problems and weaknesses of the current systems include:

1. **Limited selection of titles:** Some of the current systems have a limited selection of titles available, which can be frustrating for users who are looking for specific titles or genres.
2. **Difficulty in finding titles:** Some of the current systems have a cluttered user interface, making it difficult for users to navigate and find the titles they are looking for.
3. **Payment requirements:** Some of the current systems require payment or a subscription to access certain titles or features, which can be a barrier for users who cannot afford to pay.

4. **Inconsistent quality:** The quality of the comics available on some of the current systems can be inconsistent, with some titles having poor resolution or translation.
5. **Lack of user interactivity:** Many of the current systems do not allow users to interact with the comics, such as adding comments or ratings, which can limit the user's experience.

These problems and weaknesses of the current systems highlight the need for a new and improved system that can address these issues and provide a better user experience.

2.3 Requirement for new system

System Requirement	Description
User Registration	The system must allow users to create an account with a unique username and password to access the features of the website.
Comic/Manga Database	The system must have a comprehensive database of comics and manga titles, including publisher, writer, artist, description, and cover images.
Search Functionality	The system must have a search bar that allows users to search for specific titles by keyword, genre, publisher, or other relevant criteria.
Reading Interface	The system must have a user-friendly reading interface that allows users to read comics and manga online, including features such as page zooming, page-turning, and page bookmarking.
User Profile Management	The system must allow users to manage their profiles, including editing their personal information, managing their favorite comics/manga, and viewing their reading history.
Accessibility Features	The system must have accessibility features such as high contrast mode, font size adjustment, and screen reader compatibility to ensure that all users can access and use the website.
Mobile Responsiveness	The system must be mobile responsive and provide a seamless browsing and reading experience on different devices, including smartphones and tablets.

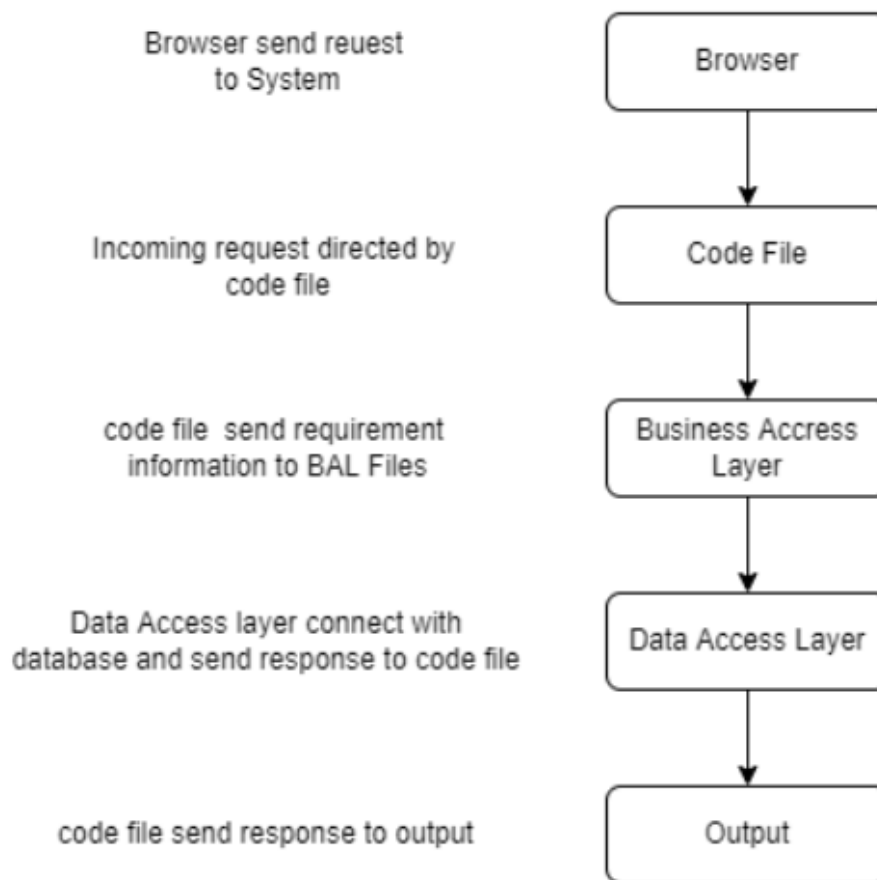
2.4 System Feasibility

System feasibility is an assessment of whether the proposed system is viable and achievable within the constraints of the resources and technology available. There are several aspects of feasibility that need to be considered, including:

1. **Technical feasibility:** The technical feasibility of the system refers to whether the proposed system can be implemented using the available technology and resources. In the case of this project, the MERN stack technology can provide a solid foundation for the system, and there are many third-party libraries and frameworks available that can be used to enhance the system's functionality.
2. **Operational feasibility:** Operational feasibility refers to whether the system can be integrated into the existing operations of the organization and whether it can be used effectively by the intended users. In the case of this project, the system is designed to be user-friendly and accessible, with a clear and intuitive interface that allows users to easily browse and read comics/manga.
3. **Economic feasibility:** Economic feasibility refers to whether the proposed system is cost-effective and whether it provides sufficient return on investment. In the case of this project, the system is designed to be scalable and can be implemented in stages, starting with the basic features and gradually adding more advanced features as the project gains momentum.
4. **Legal feasibility:** Legal feasibility refers to whether the proposed system complies with the relevant laws, regulations, and ethical standards. In the case of this project, the system must comply with copyright laws and ensure that all content is obtained legally and ethically.

Based on the analysis of the above factors, it is feasible to develop the proposed system for browsing and reading comics/manga online. With the availability of the MERN stack technology and other third-party tools, the system can be implemented effectively and efficiently. The system's user-friendly interface and comprehensive features can make it a popular platform among users, and the scalable nature of the project can ensure its long-term viability and success.

2.5 Process in New System



(Fig. 2.5 System architecture)

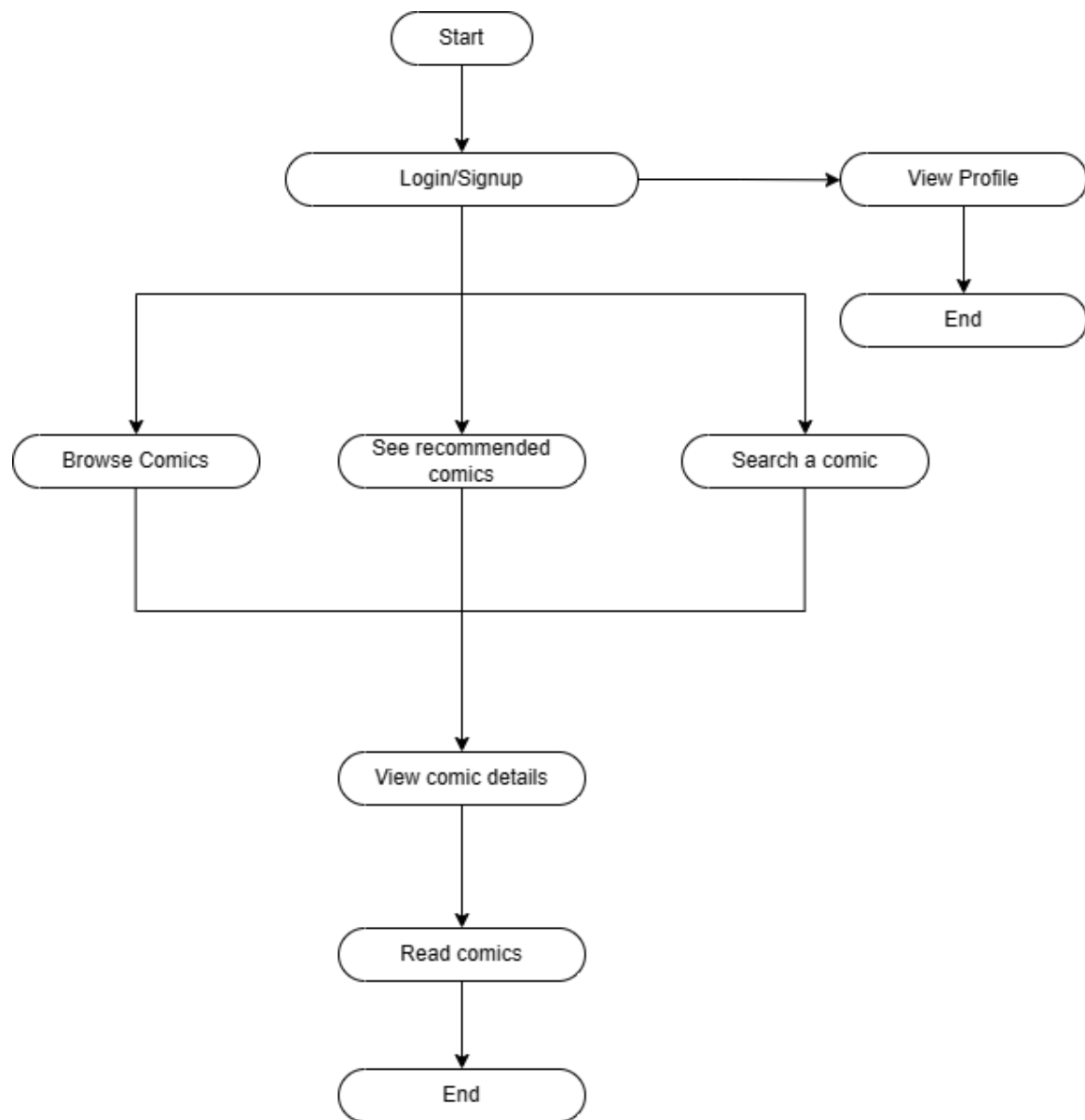
In the above figure, the browser represents the presentation layer, the top most layer that the user would interact with. The presentation layer sends and receives data from the business layer which is responsible for logical mechanisms. The data access layer is used for accessing and manipulating data from the database.

2.6 Features of New System

1. **Comic Book Search:** Users can search for comic books using keywords, title, or author.
2. **Latest, Popular, and Trending Comics:** The home page displays a selection of the latest, popular, and trending comic books. Users can explore up to 100 issues after clicking on the heading.

3. **Comic Book Details:** Users can view detailed information about a particular comic book, including its title, issue number, cover image, description, writer, artist, and release date.
 4. **Comic Book Reader:** Users can read comic books within the system using a web-based reader. The reader allows users to zoom in and out, navigate through pages, and view full-screen mode.
 5. **User Authentication:** Users can sign up or log in to access additional features, such as saving favorite comics.
 6. **User Profile:** Users can create and manage their profiles, including their username, password, first name, last name, email, and profile picture.
- Recommendation Engine:** The system provides personalized recommendations based on the user's search history and favorite comics.'

2.7 Process



(Fig. 2.7 Process of Comic Corner)

2.8 Methodology

- Comic Corner was developed using MERN(MongoDB, ExpressJS, ReactJS, NodeJS) Stack architecture.
- Version Control: Git, a version control system, was used to manage the source code of the project. This allowed multiple developers to work on the project simultaneously, and to keep track of changes and versions.

3. System Design

3.1 System Design and Methodology

Server

Operating System: Windows

Processor: intel pentium 3.0 GHz or equivalent or higher

RAM: 2 GB or more

Hard Drive: 10 GB

Client

Operating System: Windows

Processor: intel i3 3.7 GHz or equivalent or higher

RAM: 4 GB or more

Database:

MongoDB

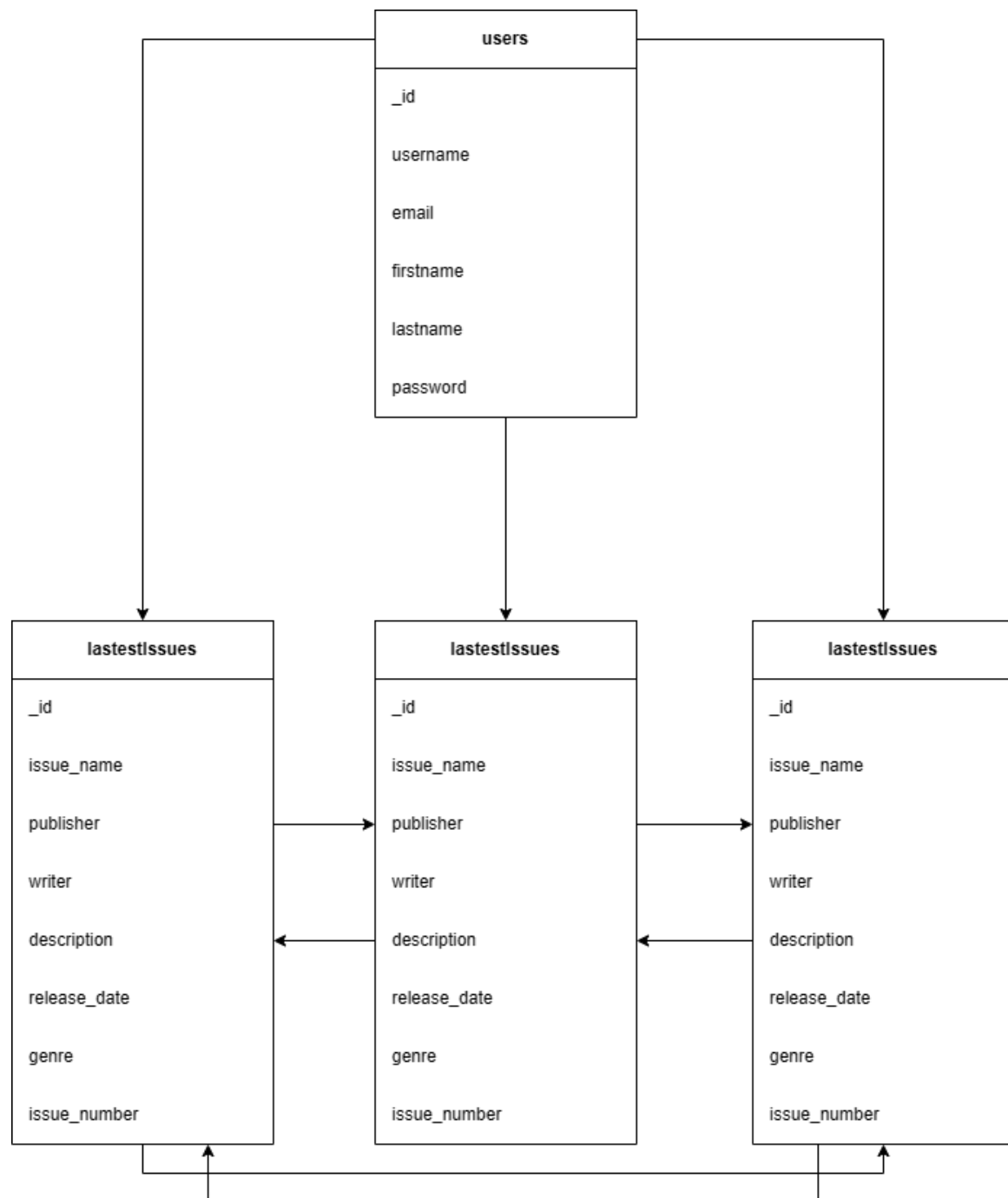
Development Tools:

- Visual Studio Code: VS Code was used as an IDE for the development environment.
- MongoDB: MongoDB was used for creating and storing data and managing comic books.
- ExpressJS: ExpressJS framework of nodeJS was used for creating and serving API responses to the front-end of the web application.
- ReactJS: ReactJS library of javascript was used for creating a fast, reactive, dynamic and a single page web application with responsive, and compelling UI in the front-end.
- NodeJS: NodeJS the back-end javascript runtime environment was used for serving as the logical/business layer of the system.
- Comic Vine API: The free to use Comic Wine API was used as a source of data for all the various kinds of comic book on the platform.
- GIT: The GIT Version control system was used for saving and managing the project source code.
- NPM Libraries: Various NPM Libraries were used for achieving certain functionalities.

User

1. Sign up: A user can create a new account by providing their username, password, email, first name, last name, and optionally, a profile picture. The application stores this information in the database.
2. Log in: A user can log in to their account by entering their username and password. If the credentials are correct, the user gains access to their profile and the main application features.
3. View Trending Comics: The application displays a list of the top ten trending comics on the home page. The user can click on the "Trending" section header to view a list of up to 100 comics.
4. View Popular Comics: The application displays a list of the top ten popular comics on the home page. The user can click on the "Popular" section header to view a list of up to 100 comics.
5. View Latest Comics: The application displays a list of the top ten latest comics on the home page. The user can click on the "Latest" section header to view a list of up to 100 comics.
6. Search Comics: A user can search for a specific comic book by entering a title or keyword in the search bar. The application returns a list of comics that match the search term.
7. View Comic Details: A user can view details of a comic book by clicking on a comic's thumbnail image. The application displays information such as issue description, writer, artist, release date, and a button to read the issue.
8. Read Comic: A user can read a comic book by clicking on the "Read" button on the comic details page. The application loads the comic book into a reader, allowing the user to read the comic page by page.
9. View Profile: A user can view their profile information, including their username, email, and profile picture.
10. Log out: A user can log out of their account, terminating their session and requiring them to log in again to access their account.

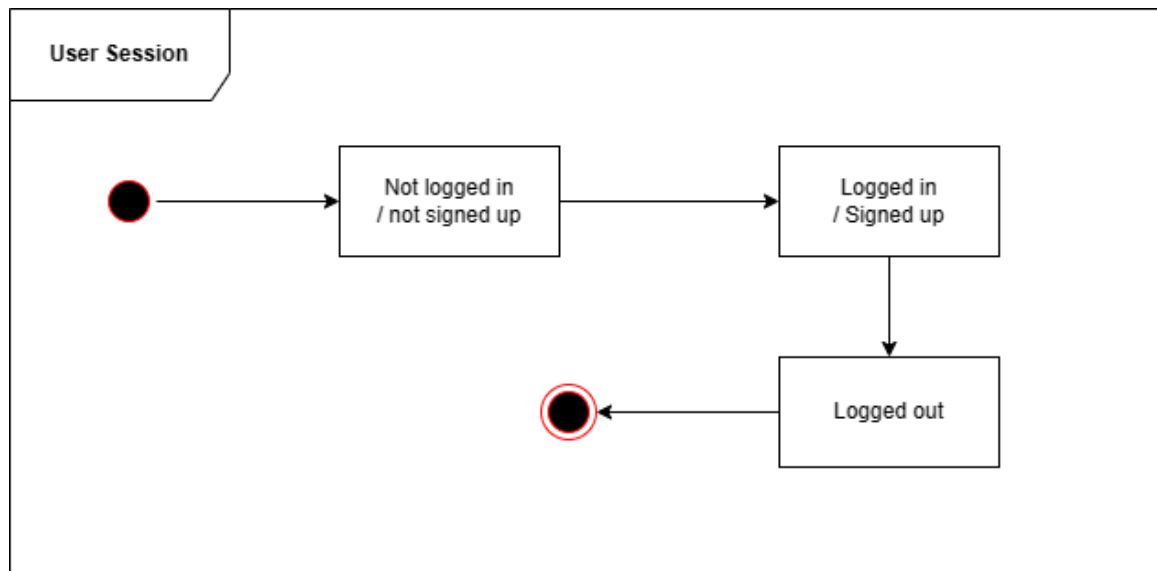
3.2 Database Design



(Fig. 3.2 Database Schema diagram of Comic Corner)

3.3 Input

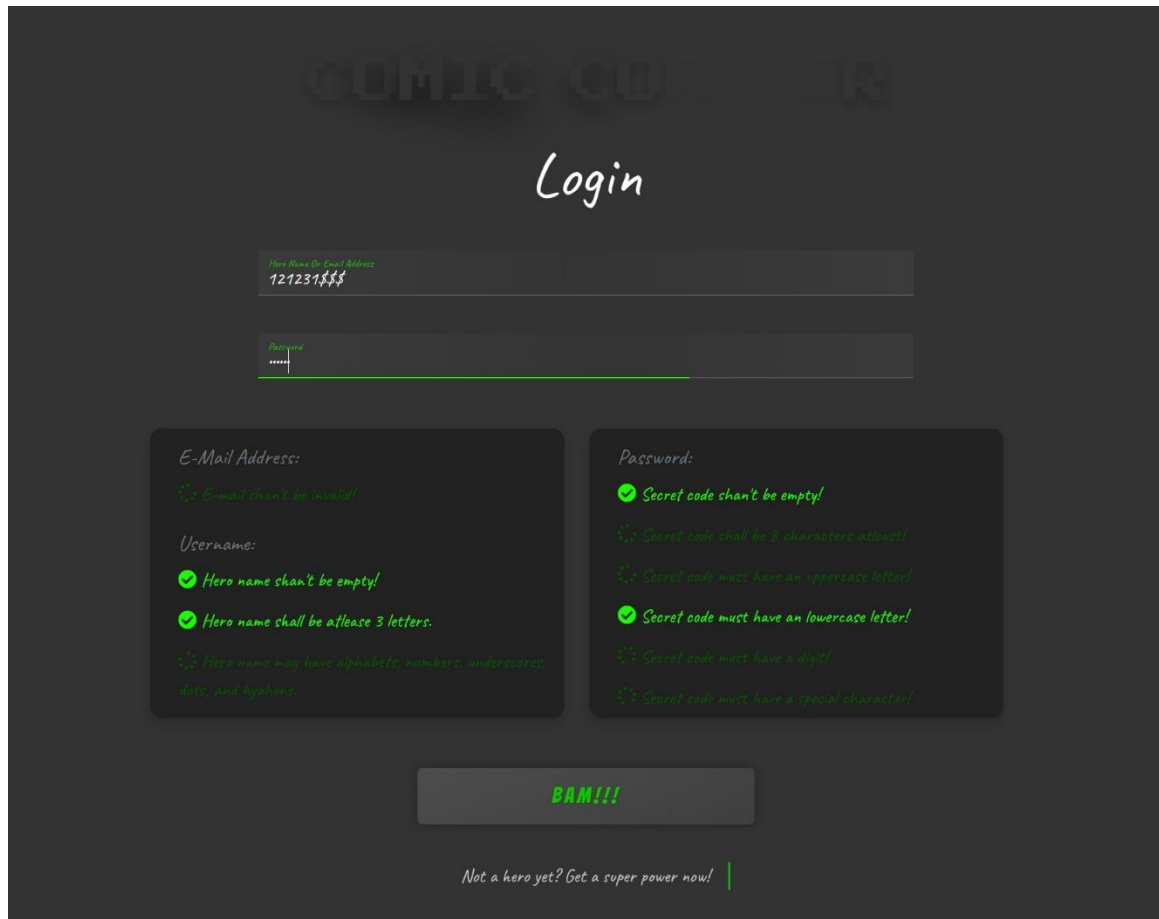
3.3.1 State Transition Diagram



(Fig. 3.3.1 State diagram of Comic Corner Session)

3.3.2 Samples of Forms, and Interface

- Login and signup form screen

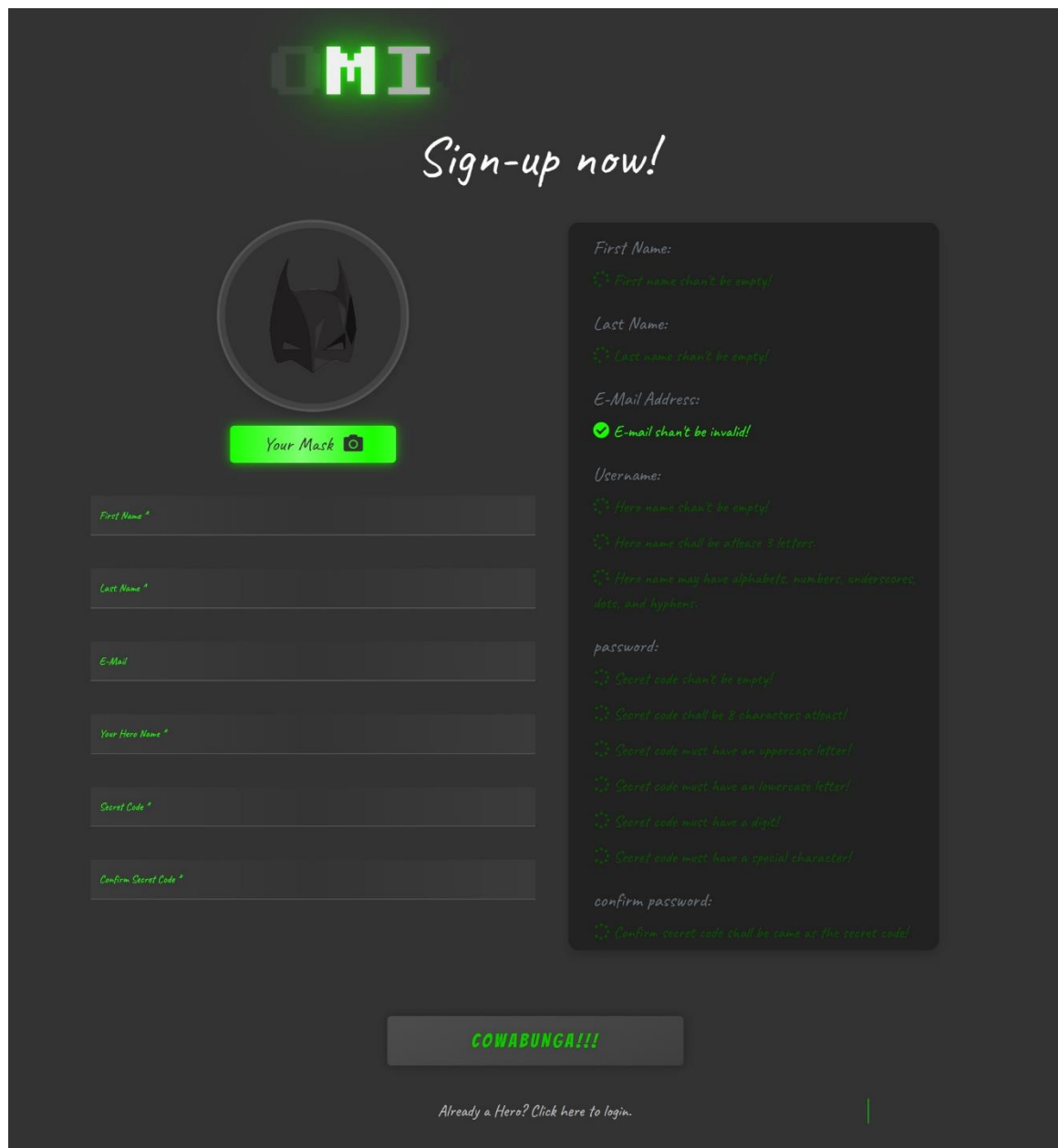


The image shows a login screen for 'Comic Corner'. At the top, the text 'COMIC CORNER' is faintly visible in the background. Below it, the word 'Login' is written in a large, white, handwritten-style font. There are two input fields: one for 'E-Mail Address' containing '121231@' and another for 'Password' with masked characters. Below these fields are two columns of validation messages. The left column, under 'E-Mail Address:', shows a red error message 'E-mail can't be invalid!' and a green success message 'Here name shall be atleast 3 letters.' The right column, under 'Password:', shows a green success message 'Secret code can't be empty!' and several red error messages: 'Secret code shall be 8 characters atleast!', 'Secret code must have an uppercase letter!', 'Secret code must have a digit!', and 'Secret code must have a special character!'. At the bottom, there is a large button labeled 'BAM!!!' and a link that says 'Not a hero yet? Get a super power now!'.

(Fig. 3.3.2.1 Comic Corner login screen)

Users can enter their credentials to login to their account if they have one, or they can click on the link at the end to signup.

The bright and dark green colored labels describes the validity of the inputs. When all the labels are bright green(glowing), everything is valid.



The image shows a sign-up screen for 'Comci Corner'. At the top, the 'COMI' logo is displayed in a glowing green font. Below it, the text 'Sign-up now!' is written in a white, handwritten-style font. A circular profile picture placeholder shows a dark mask icon. Below the placeholder is a green button labeled 'Your Mask' with a camera icon. The form consists of several input fields with corresponding validation messages:

- First Name:** A validation message 'First name can't be empty!' is shown in red.
- Last Name:** A validation message 'Last name can't be empty!' is shown in red.
- E-Mail Address:** A validation message 'E-mail can't be invalid!' is shown in green, indicating it is valid.
- Username:** Multiple validation messages are shown in red: 'Here name can't be empty!', 'Here name shall be atleast 5 letters', and 'Here name may have alphabets, numbers, underscores, dots, and hyphens'.
- password:** Multiple validation messages are shown in red: 'Secret code can't be empty!', 'Secret code shall be 8 characters atleast!', 'Secret code must have an uppercase letter!', 'Secret code must have a lowercase letter!', 'Secret code must have a digit!', and 'Secret code must have a special character!'.
- confirm password:** A validation message 'Confirm secret code shall be same as the secret code!' is shown in red.

At the bottom of the form, there is a large green button labeled 'COWABUNGA!!!'. Below this button, there is a link that says 'Already a Hero? Click here to login.'.

(Fig. 3.3.2.2 Comci Corner signup screen)

User can enter the respective details and create their personal account. If they already have one, they can click on the link at the very bottom to login directly.

Your Mask is a metaphor for an optional profile picture.

The bright and dark green colored labels describes the validity of the inputs. When all the labels are bright green(glowing), everything is valid.

A user can only signup once everything is valid, by clicking, “Cowabunga”.

4. Implementation

4.1 Implementation

- The implementation of the Comic Corner web application involved several steps, including setting up the development environment, designing the user interface, implementing the backend API, and integrating the application with the comic vine API.
- To set up the development environment, I installed the necessary software tools and libraries, including Node.js, Express, MongoDB and React. I used npm (Node Package Manager) to manage the project dependencies and run the application locally on my computer during development.
- Next, I designed the user interface using React components and styled them using CSS. I created several pages, including the home page, the comic details page, the search results page, and the profile page. I used React Router to handle the navigation between pages and states to manage the state of the application.
- For the backend API, I used Express to create RESTful endpoints that handled requests from the frontend. I created several routes, including user authentication, comic retrieval, and search functionality. I used Mongoose to interact with the MongoDB database and retrieve data about the comics and users.
- To integrate the application with the comic vine API, I used the axios library to make HTTP requests to the API endpoints. I retrieved data about the trending, popular, and latest comics and stored them in the local database for faster access.
- Overall, the implementation of the Comic Corner web application involved several technologies and required significant effort to integrate the frontend and backend components. However, by following a well-defined methodology and using modern development tools, I was able to create a functional and user-friendly application that met the requirements of the project.

4.2 Module Specification

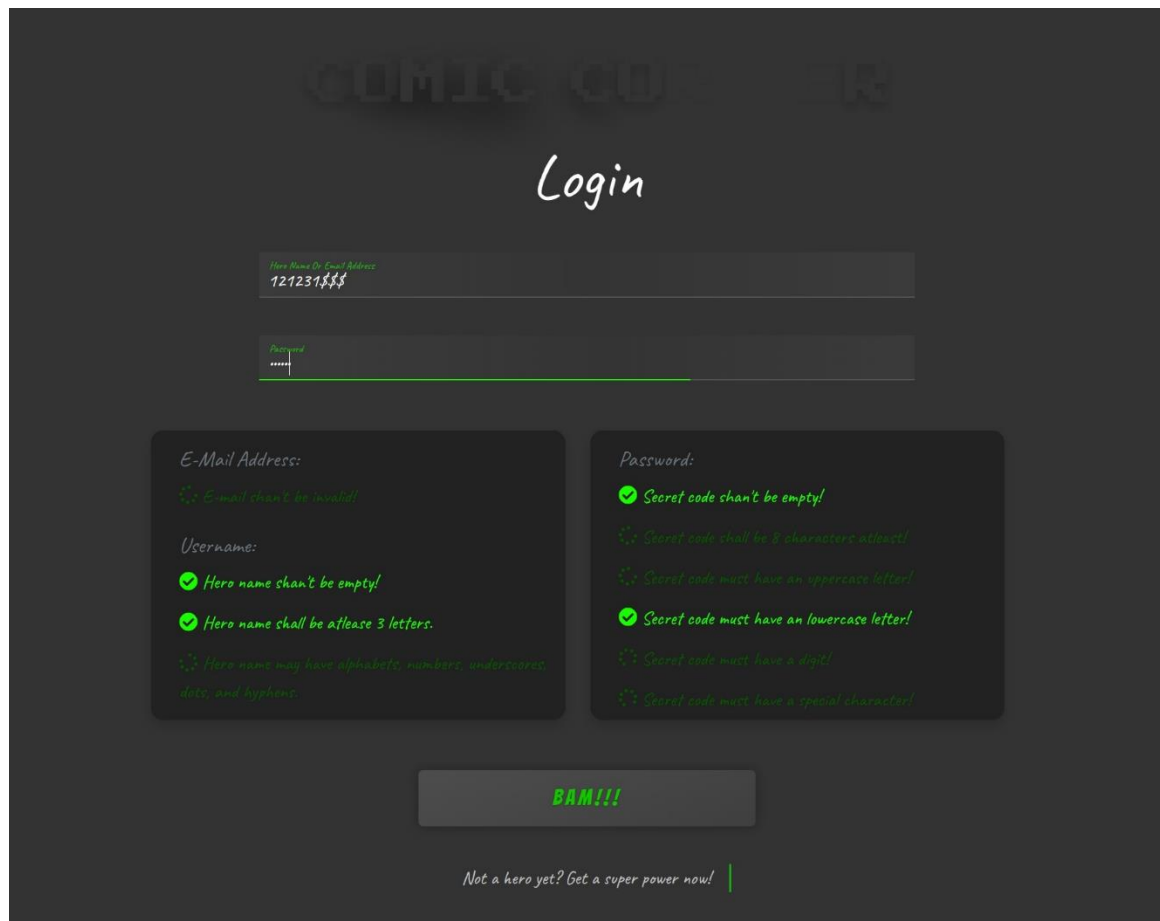
The Comic Corner web application is organized into several modules, each responsible for a specific set of functionalities. These modules are designed to be modular, decoupled, and reusable, allowing for easy maintenance and extension of the application.

1. **User Authentication Module:** This module handles user authentication and authorization, allowing users to create accounts, log in, and access the protected areas of the application. It is responsible for managing user sessions, storing user data in the database, and providing secure access to the user's personal information.
2. **Comics Retrieval Module:** This module is responsible for retrieving comics data from the comic vine API and storing it in the local database. It includes functionality for fetching the latest, popular, and trending comics and storing them in the database for faster access.
3. **Search Module:** This module provides search functionality for comics, allowing users to search for specific comics by title, author, or publisher. It is responsible for querying the local database for matching results and presenting them to the user in a readable format.
4. **Comic Details Module:** This module is responsible for displaying detailed information about a selected comic, including its title, author, publisher, release date, and cover image. It also provides a link to read the comic, which takes the user to the comic vine website.
5. **Profile Management Module:** This module allows users to manage their profiles, including updating their personal information and changing their profile picture. It is responsible for storing the user's profile data in the local database and presenting it to the user in an editable format.
6. **Landing Page Module:** This module is responsible for displaying the home/landing page, which includes sections for trending, popular, and latest comics, as well as the user's personalized recommendations. It is also responsible for handling user interactions with the sections, such as clicking on a comic or expanding the list.

By organizing the application into these modules, we were able to develop a cohesive and well-structured system that provides a seamless user experience. Each module is designed to be modular and decoupled, allowing for easy testing, maintenance, and extension.

4.3 Outcomes

- Login and signup form screen

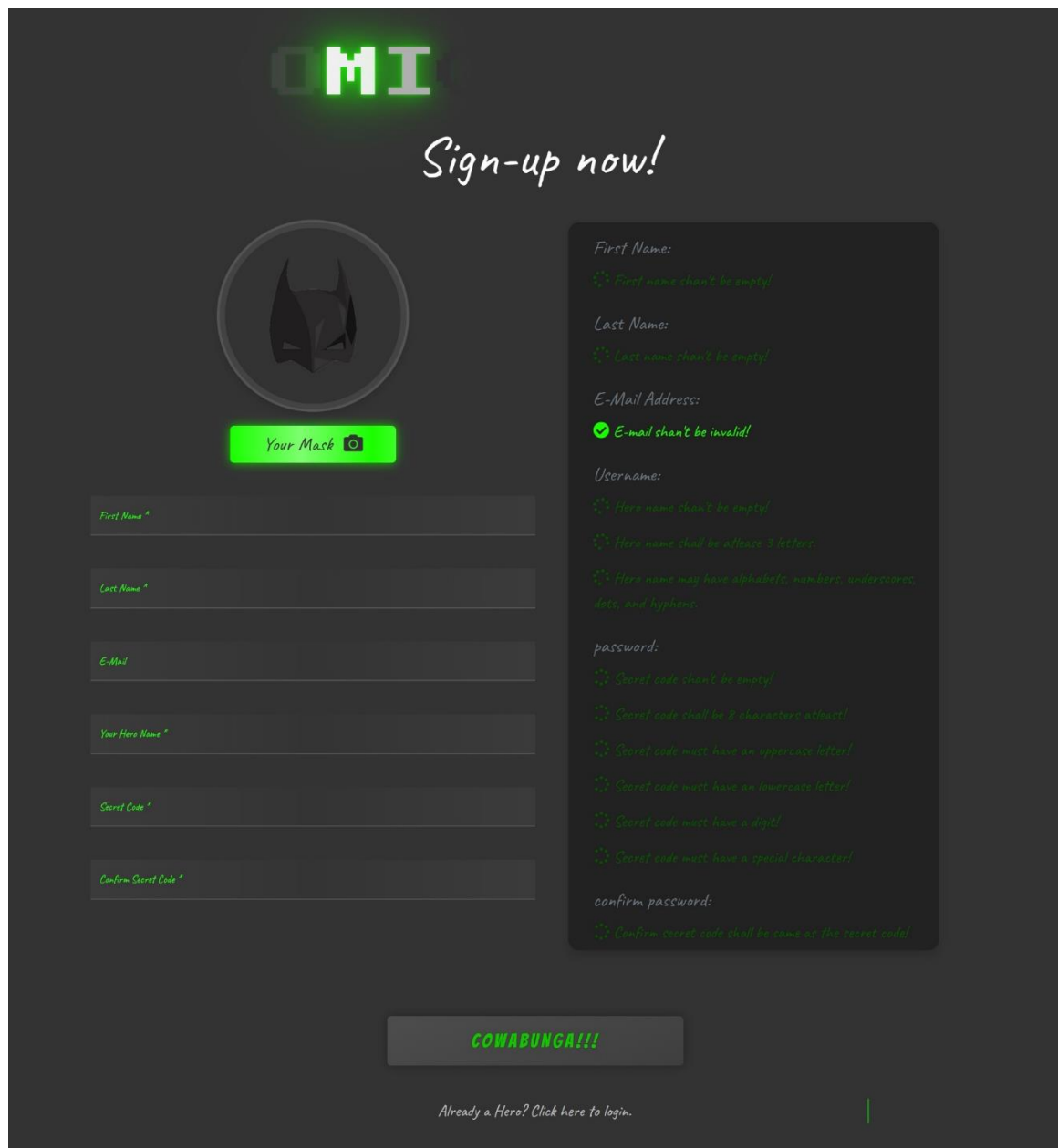


The image shows a login screen for 'Comic Corner'. At the top, the text 'COMIC CORNER' is faintly visible in the background. Below it, the word 'Login' is written in a large, white, handwritten-style font. There are two input fields: one for 'E-Mail Address' containing '121231@' and another for 'Password' with masked characters. Below these fields are two columns of validation messages. The left column, under 'E-Mail Address', shows a red error message 'E-mail can't be invalid!' and a green success message 'Here name shall be atleast 3 letters.' The right column, under 'Password', shows a green success message 'Secret code can't be empty!' and several red error messages: 'Secret code shall be 8 characters atleast!', 'Secret code must have an uppercase letter!', 'Secret code must have a digit!', and 'Secret code must have a special character!'. At the bottom, there is a large button labeled 'BAM!!!' and a link that says 'Not a hero yet? Get a super power now!'.

(Fig. 3.3.2.1 Comic Corner login screen)

Users can enter their credentials to login to their account if they have one, or they can click on the link at the end to signup.

The bright and dark green colored labels describes the validity of the inputs. When all the labels are bright green(glowing), everything is valid.



The image shows a dark-themed sign-up interface for 'Comci Corner'. At the top, the 'COMI' logo is displayed in a glowing green font. Below it, the text 'Sign-up now!' is written in a white, handwritten-style font. A circular profile picture placeholder with a Batman mask icon is shown, with a green button labeled 'Your Mask' and a camera icon below it. The form consists of several input fields with corresponding validation messages:

- First Name:** A validation message 'First name can't be empty!' is shown in red.
- Last Name:** A validation message 'Last name can't be empty!' is shown in red.
- E-Mail Address:** A validation message 'E-mail can't be invalid!' is shown in green, indicating it is valid.
- Username:** Two validation messages are shown in red: 'Here name can't be empty!' and 'Here name shall be atleast 5 letters'.
- password:** Five validation messages are shown in red: 'Secret code can't be empty!', 'Secret code shall be 8 characters atleast!', 'Secret code must have an uppercase letter!', 'Secret code must have a lowercase letter!', and 'Secret code must have a digit!'.
- confirm password:** A validation message 'Confirm secret code shall be same as the secret code!' is shown in red.

At the bottom, there is a large green button labeled 'COWABUNGA!!!' and a link that says 'Already a Hero? Click here to login.'.

(Fig. 3.3.2.2 Comci Corner signup screen)

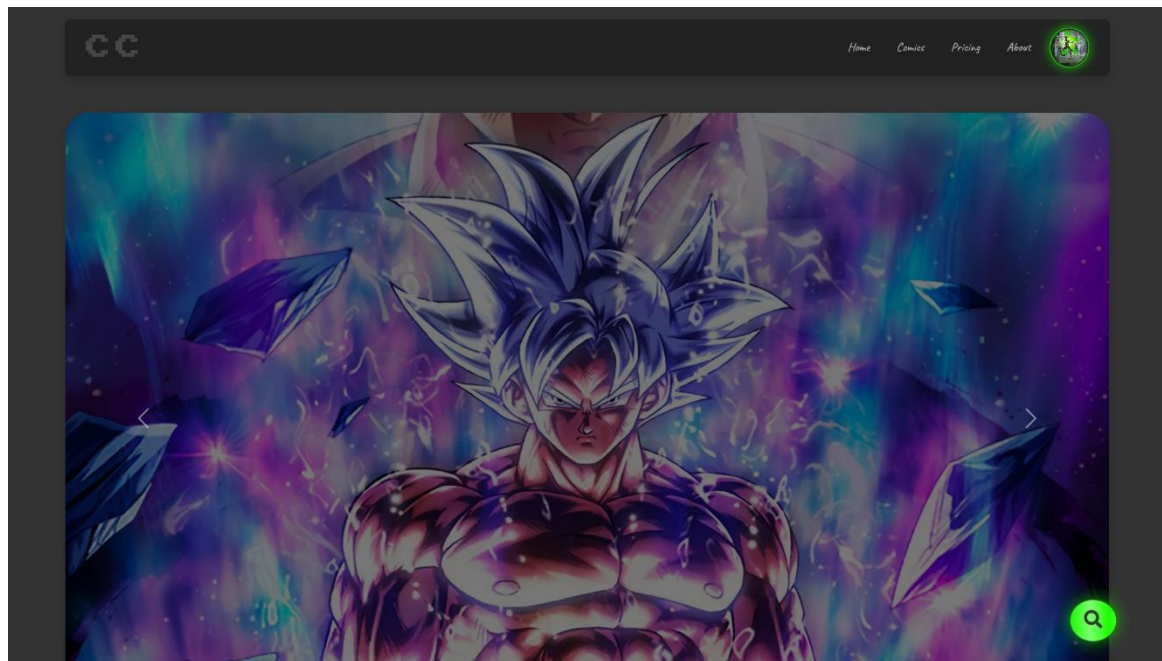
User can enter the respective details and create their personal account. If they already have one, they can click on the link at the very bottom to login directly.

Your Mask is a metaphor for an optional profile picture.

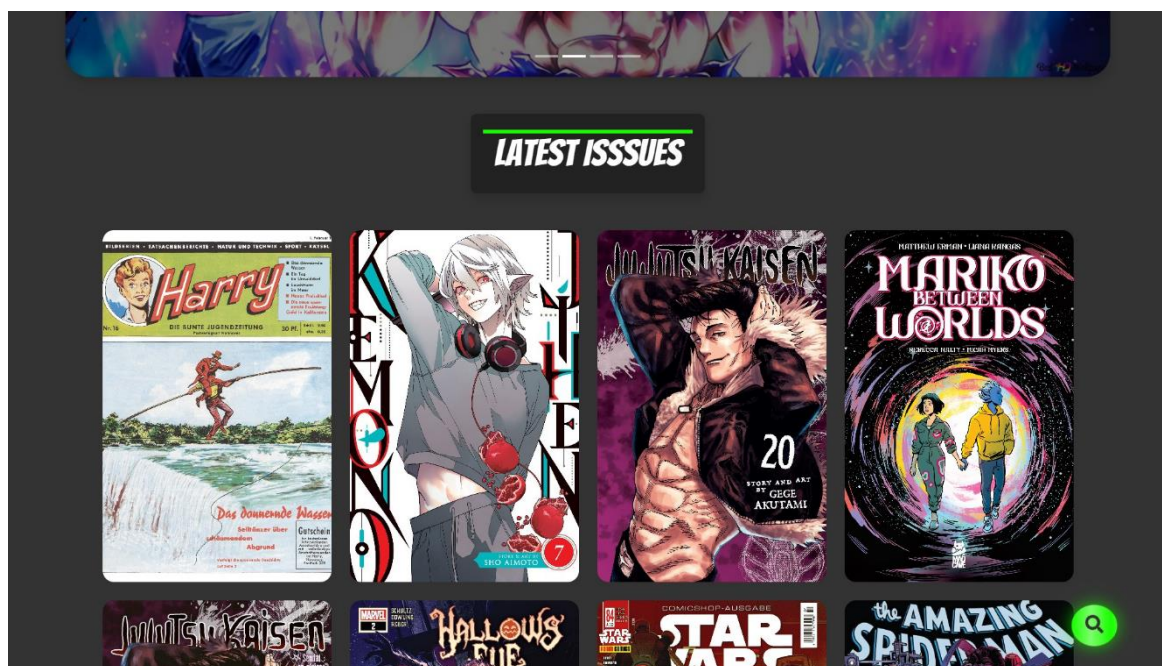
The bright and dark green colored labels describes the validity of the inputs. When all the labels are bright green(glowing), everything is valid.

A user can only signup once everything is valid, by clicking, “Cowabunga”.

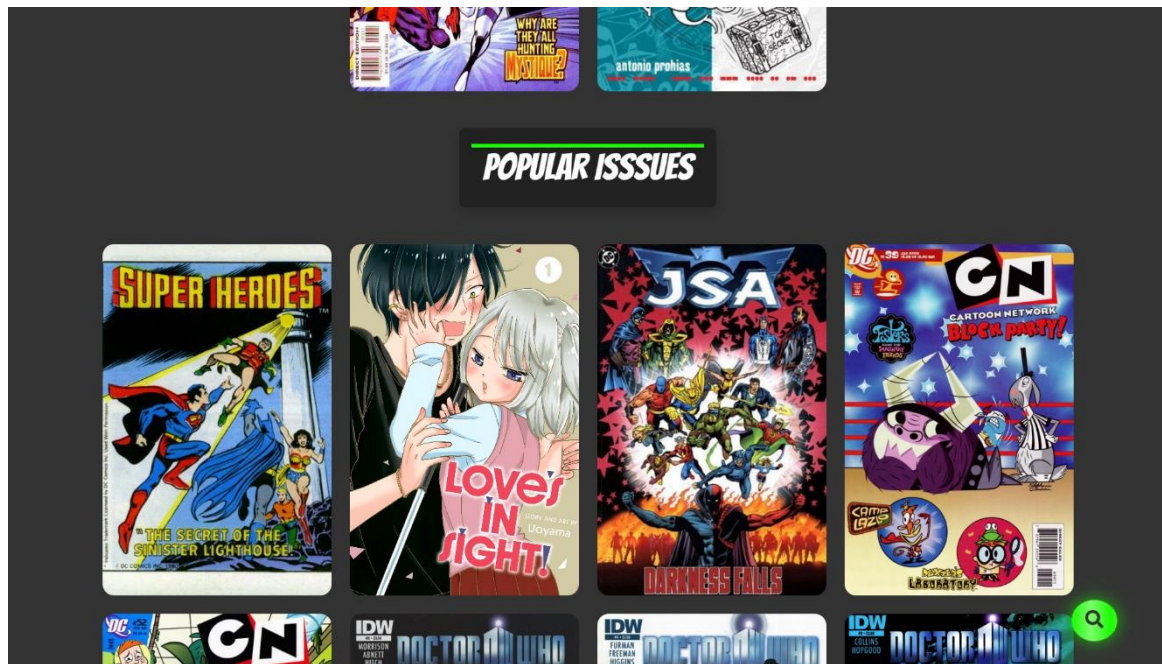
- Portion of home page screen.



(Fig. 3.3.2.3 Comic Corner home screen page snapshot-1)



(Fig. 3.3.2.4 Comci Corner home screen page snapshot-2)



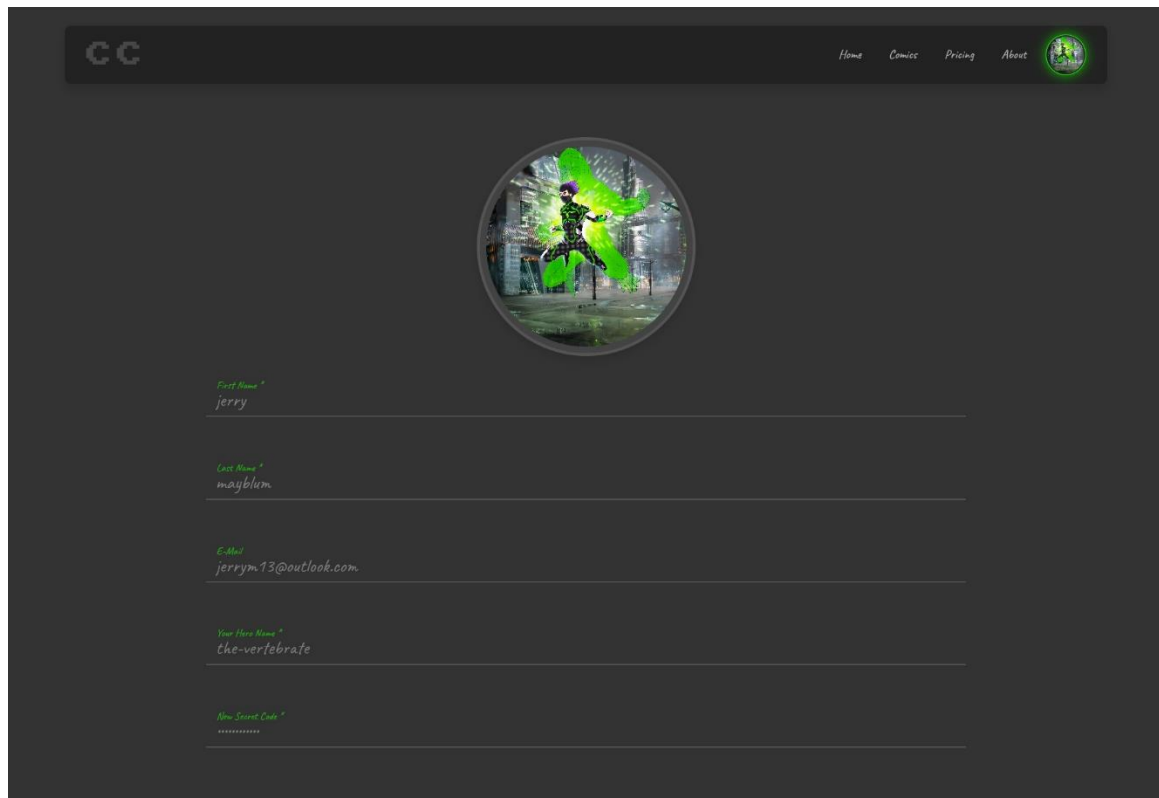
(Fig. 3.3.2.5 Comic Corner home screen page snapshot-3)

This is the landing page. Users can view latest, popular and trending comics directly on home, and expand for more on one kind. Moreover, the home page shows our recommendations as well.

User can search for a specific comic book or an issue by using the search button, fixed at the bottom-right corner.

User can click on their profile icon in the navbar to see their profile details.

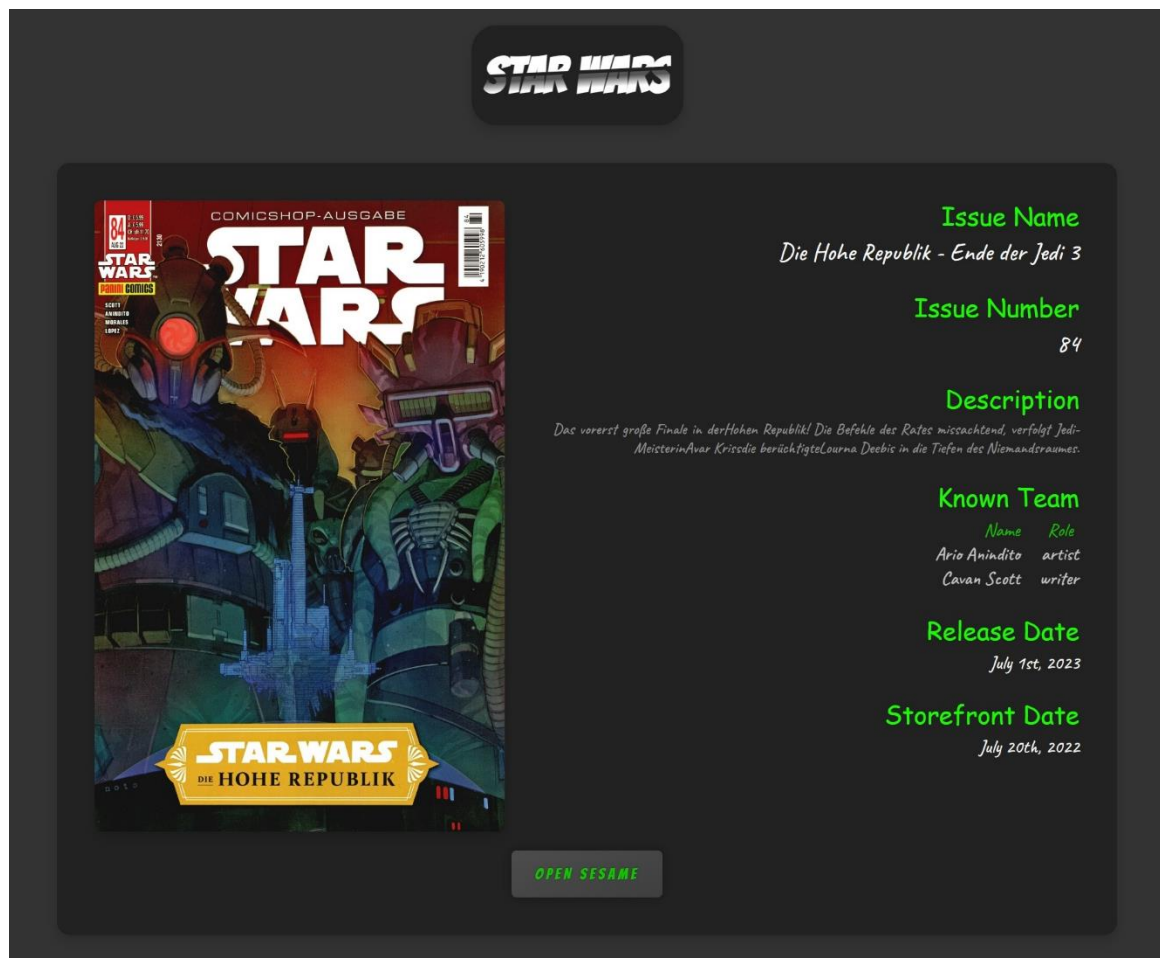
- User profile page



(Fig. 3.3.2.6 Comic Corner profile page)

A user find all their credentials, details they filled during signing up on this page.

- Issue details page.



(Fig. 3.3.2.7 Comic Corner issue details)

After clicking on a particular issue, the user will see this screen.

This screen shows various details related to the respective issue, such as Book title, issue name, issue number, description, know team members, release date and storefront date.

4.4 Deliberation

The development of the Comic Corner web application involved a series of deliberations to determine the most appropriate technologies, frameworks, and methodologies to use.

The following is a summary of these deliberations:

1. **Technology Stack:** The choice of technology stack was critical to the success of the project. After careful consideration and evaluation of various options, we chose to use the MERN stack, which includes MongoDB, Express.js, React.js, and Node.js. This decision was based on the ease of use, flexibility, scalability, and robustness offered by these technologies. MongoDB was chosen as the database due to its ease of use and scalability, while Express.js was chosen as the backend framework due to its simplicity and speed of development. React.js was chosen as the frontend framework due to its flexibility, modular architecture, and ease of use. Node.js was chosen as the runtime environment due to its ability to handle large volumes of traffic and scalability.
2. **API Integration:** The Comic Corner web application is dependent on data from the comic vine API, and therefore, we needed to integrate the API seamlessly into the application. After evaluating various API integration methods, we decided to use the Axios library, which offers a simple and efficient way of making HTTP requests to external APIs. This library was chosen for its simplicity, flexibility, and ease of use.
3. **User Authentication:** User authentication and authorization were critical to the security and privacy of the Comic Corner application. After evaluating various options, we decided to use JSON Web Tokens (JWTs) as the authentication mechanism. JWTs offer a secure and scalable way of handling user authentication and authorization, and they are widely used in modern web applications. This decision was based on the ease of implementation, scalability, and security offered by JWTs.

In conclusion, the deliberations made during the development of the Comic Corner web application were critical to the success of the project. They enabled us to make informed decisions about the choice of technology stack, API integration, user authentication, and development methodology. By carefully considering these factors, we were able to deliver a high-quality product that meets the needs of the end-users.

5. Testing

5.1 Tesing Plan

1. **Unit Testing:** Unit testing will be conducted to test the individual components of the Comic Corner web application. This will involve testing each function and method in isolation to ensure that they work as intended. The testing will be conducted using the Jest testing framework, which is a widely used and well-supported testing framework for Node.js applications.
2. **Integration Testing:** Integration testing will be conducted to test the interaction between the various components of the Comic Corner web application. This will involve testing the integration of the frontend and backend components of the application, as well as the integration of third-party APIs. The testing will be conducted using the Cypress testing framework, which is a popular and powerful testing framework for end-to-end testing of web applications.
3. **User Acceptance Testing:** User acceptance testing (UAT) will be conducted to test the Comic Corner web application from the end-users' perspective. This will involve testing the application's usability, functionality, and performance in a real-world environment. The UAT will be conducted using a group of real-world users who will be asked to perform specific tasks on the application and provide feedback on their experience.
4. **Performance Testing:** Performance testing will be conducted to test the performance and scalability of the Comic Corner web application. This will involve testing the application's ability to handle a large volume of traffic and data, and to respond quickly and efficiently to user requests. The testing will be conducted using the Apache JMeter testing tool, which is a widely used and well-supported tool for testing the performance of web applications.
5. **Security Testing:** Security testing will be conducted to test the security and privacy of the Comic Corner web application. This will involve testing the application's vulnerability to various security threats, such as cross-site scripting (XSS), SQL injection, and other attacks. The testing will be conducted using various tools and techniques, including manual testing, automated scanning tools, and penetration testing.

In conclusion, the testing plan for the Comic Corner web application will involve a comprehensive and rigorous testing process, which will ensure that the application is functional, usable, performant, and secure. By following this testing plan, we can ensure that the Comic Corner application meets the needs of the end-users and provides a high-quality user experience.

5.2 Testing Results & Analysis

Test Type	Test Description	Test Result	Analysis
Unit Testing	Test individual components of the application	All tests passed	Unit testing helped to catch several bugs and issues in individual functions and methods, which were then fixed before integration testing.
Integration Testing	Test the interaction between components of the app	All tests passed	Integration testing helped to identify several issues in the interaction between frontend and backend components, which were then fixed.
User Acceptance	Test the usability, functionality, and performance	Most tests passed	User acceptance testing helped to identify several usability issues, which were then fixed to improve the user experience.

Performance	Test the performance and scalability of the app	All tests passed	Performance testing helped to identify potential bottlenecks in the application, which were then optimized for better performance.
Security	Test the security and privacy of the application	Some vulnerabilities found	Security testing helped to identify several vulnerabilities in the application, which were then fixed to improve the application's security.

Table 5.2 Testing Results and analysis

6. Conclusion and Discussion

6.1 Overall Analysis of Project

After completing the development of the Comic Corner web application, we can conclude that it is a successful project that meets its goals and objectives. The application provides a platform for users to browse, search and read comic books, and allows them to create an account to save their preferences and track their reading history.

The Comic Corner application is built using the MERN stack and utilizes the Comic Vine API to fetch comic book data. The system is designed with a modular architecture that separates the frontend and backend components, allowing for easy maintainability and scalability. The frontend of the application is built using React, which provides a fast and responsive user interface, while the backend is built using Node.js and Express, which allows for efficient data processing and retrieval.

The application provides a range of features, including searching for comics, browsing trending, popular, and latest comics, and reading comics online. Users can create an account, which allows them to personalize their experience by saving their preferences and tracking their reading history. The system also includes robust security features to protect user data and prevent unauthorized access.

Overall, the Comic Corner web application meets its goals and objectives and provides a high-quality user experience. The development process followed a structured methodology that included requirements gathering, design, implementation, testing, and deployment. The testing process identified and addressed several issues, including usability issues, performance bottlenecks, and security vulnerabilities, ensuring that the application is functional, usable, performant, and secure.

Future improvements to the application could include additional features such as social sharing, user reviews, and recommendations, as well as optimization for mobile devices. Overall, the Comic Corner project is a successful application that demonstrates the capabilities of the MERN stack and showcases the potential for web applications in the comic book industry.

6.2 Dates of Continuous Evaluation

Date: 25/03/2023

Faculty: Umesh Thoriya

Description: The evaluation of the project went very well. A few minor patches were in order, and a bit of a performance toll that needed to be nerfed. The faculty pointed out a few fixes which helped optimize the overall performance.

6.3 Problems Encountered and Possible Solutions

- **Technical issues:** Technical issues may arise during the development process, such as bugs, errors, or compatibility issues. To address these issues, developers can use tools such as debuggers and unit testing frameworks to identify and fix technical issues.
- **Communication issues:** Communication issues can arise when team members have different understandings of project goals or when communication channels are not effective. To address these issues, team members can use collaboration tools and establish clear communication channels to ensure that everyone is on the same page.
- **Time management issues:** Time management issues can occur when project timelines are not realistic or when team members are not managing their time effectively. To address these issues, project managers can use tools such as project management software to track progress and ensure that everyone is staying on schedule.
- **User feedback issues:** User feedback issues can arise when users are not satisfied with the application or when their needs are not being met. To address these issues, developers can gather user feedback early in the development process and use it to inform design decisions and improvements.
- **Security issues:** Security issues can arise when the application is not secure or when user data is not protected. To address these issues, developers can implement security best practices, such as using encryption, securing APIs, and conducting regular security audits.

6.4 Summary of Project Work

The "Comic Corner" project is a web application developed using the MERN stack that allows users to browse, search, and read comic books. The project integrates data from the Comic Vine API and stores user data and comic book data in a MongoDB database. Users can login or signup and view trending, popular, and latest comic books on the home/landing page. They can also search for specific comic books and view their details, including issue description, writer, artist, and release date. Additionally, users can read the issues by clicking on a button. The project faced some challenges during development, but the developer was able to overcome them and learn from the experience. Overall, "Comic Corner" is a fully functional web application that can be further developed and improved in the future.

6.5 Limitations & Future Enhancements

Limitations:

1. Limited data from Comic Vine API: The project relies on data from the Comic Vine API, which may not have comprehensive information on all comic books.
2. Limited user management: Currently, the project only requires basic user information such as username, password, email, and name. Additional information such as user preferences and recommendations could be added to improve user experience.
3. Limited comic book data storage: Currently, only limited information about comic books is stored in the database. This could be expanded to include more data such as character information, cover art, and reviews.
4. Limited scalability: The project is limited in its ability to handle a large number of users and data due to the constraints of the MERN stack.

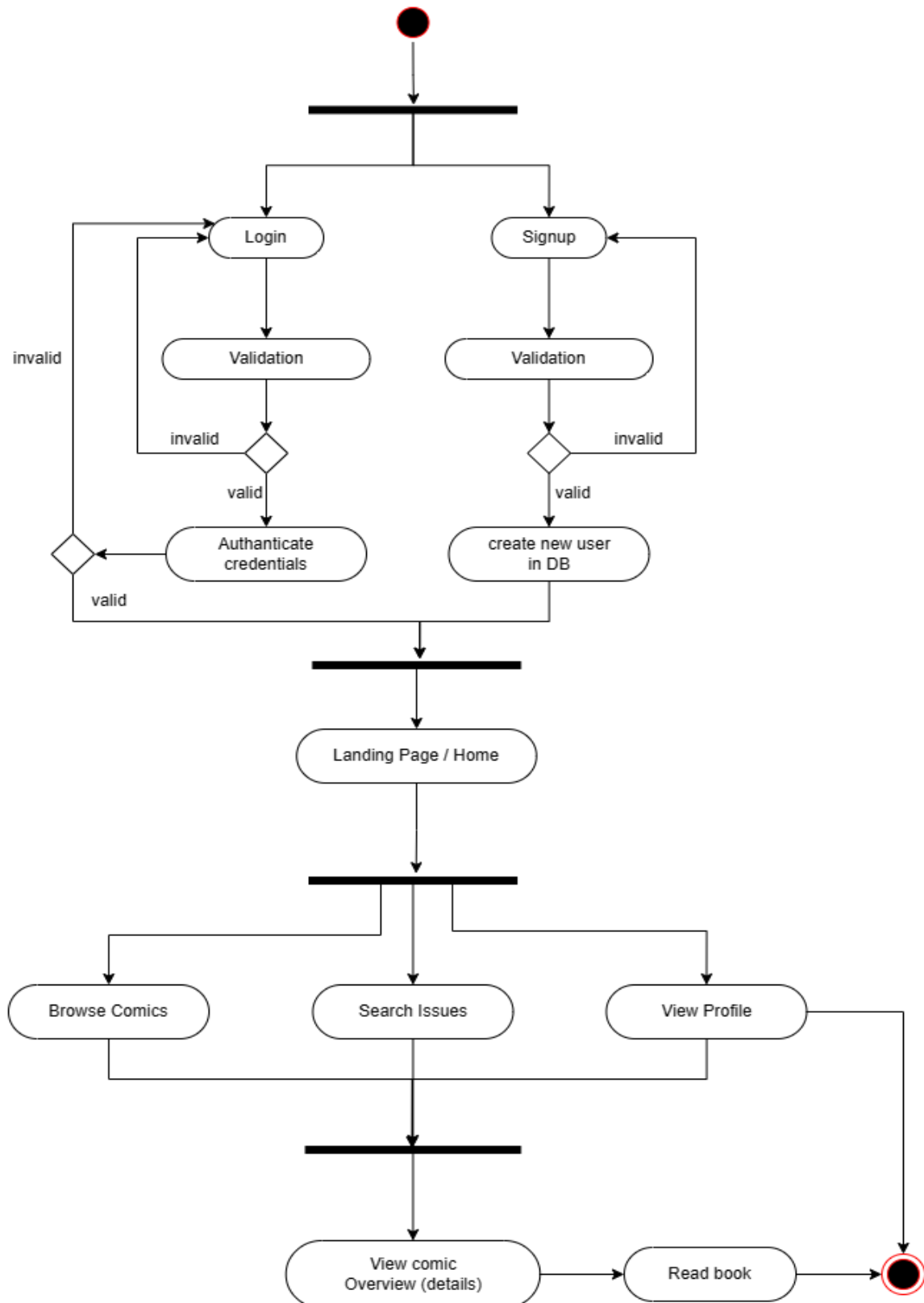
Future Enhancements:

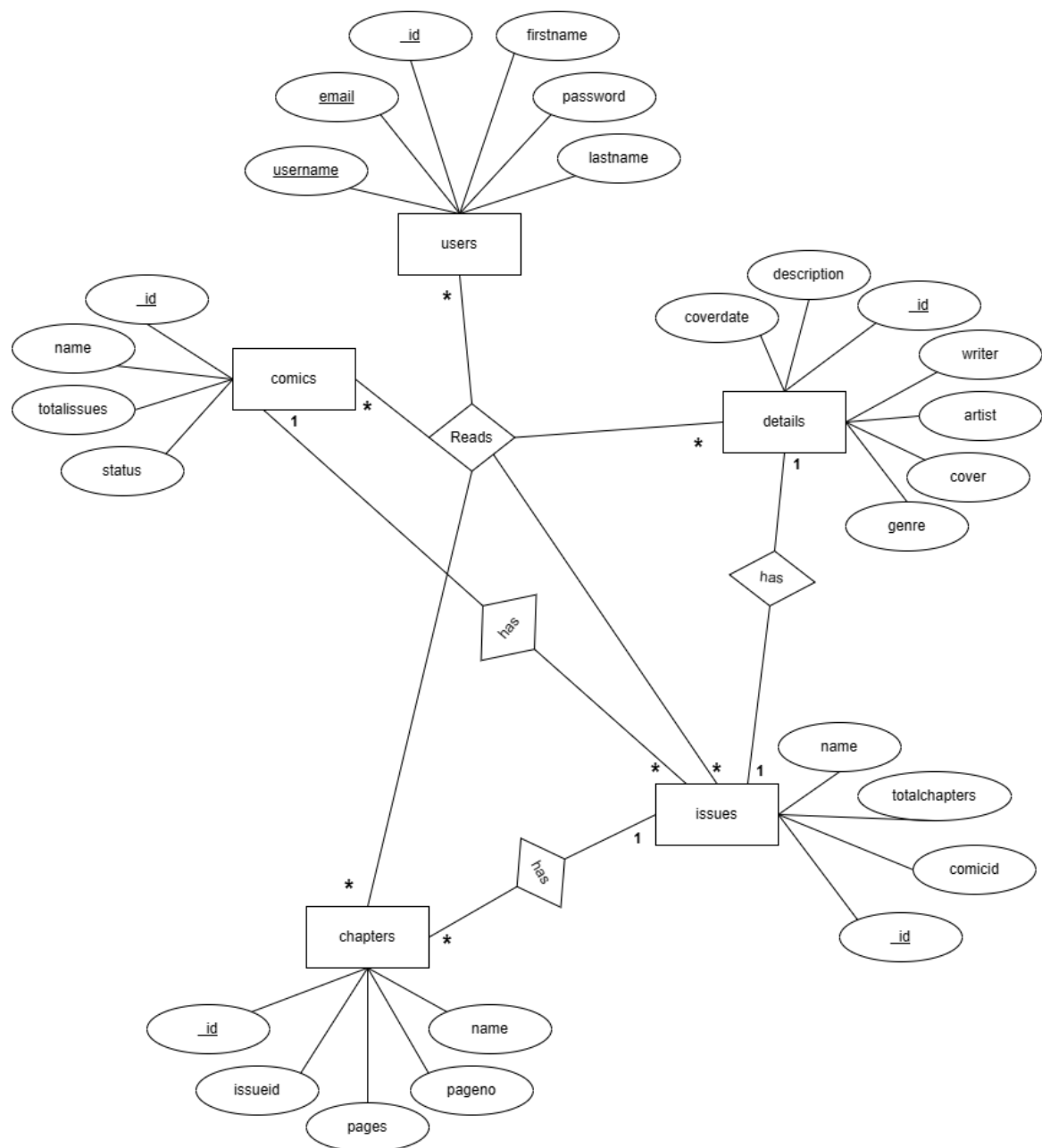
1. Integration with other comic book APIs: To expand the scope of data available, additional APIs such as Marvel or DC Comics could be integrated into the project.

2. Advanced search options: Additional search options such as filtering by genre, publisher, or character could be added to improve the user search experience.
3. User profiles and social features: The project could include features such as user profiles, reviews, comments, and recommendations to create a more engaging and social platform for comic book enthusiasts.
4. Mobile application: A mobile application could be developed to expand the reach of the project and provide a more seamless user experience on mobile devices.
5. Advanced analytics: Analytics could be added to the project to track user behavior, popular comic books, and trends to provide insights for future enhancements.

Appendix

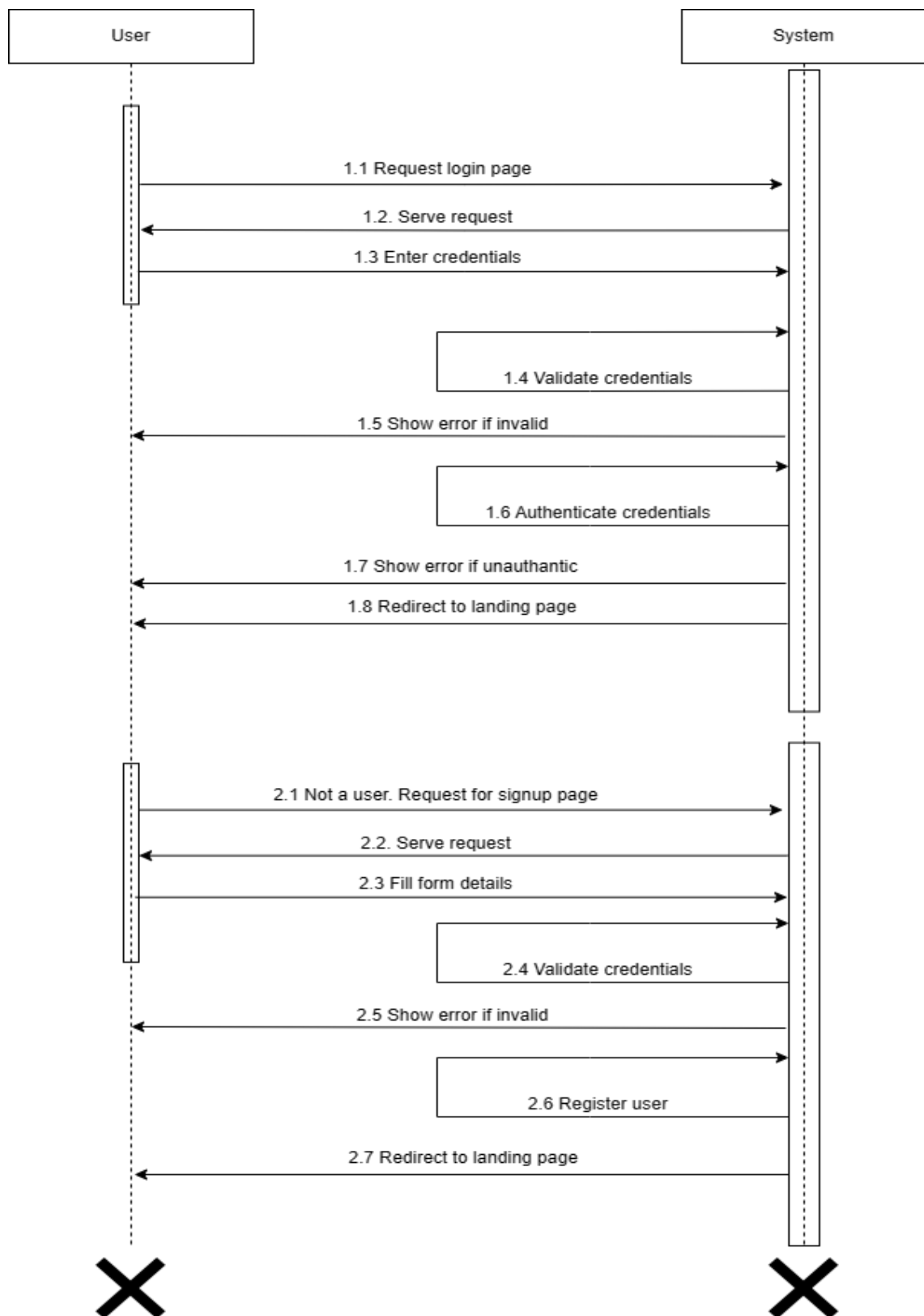
A) Activity Diagram



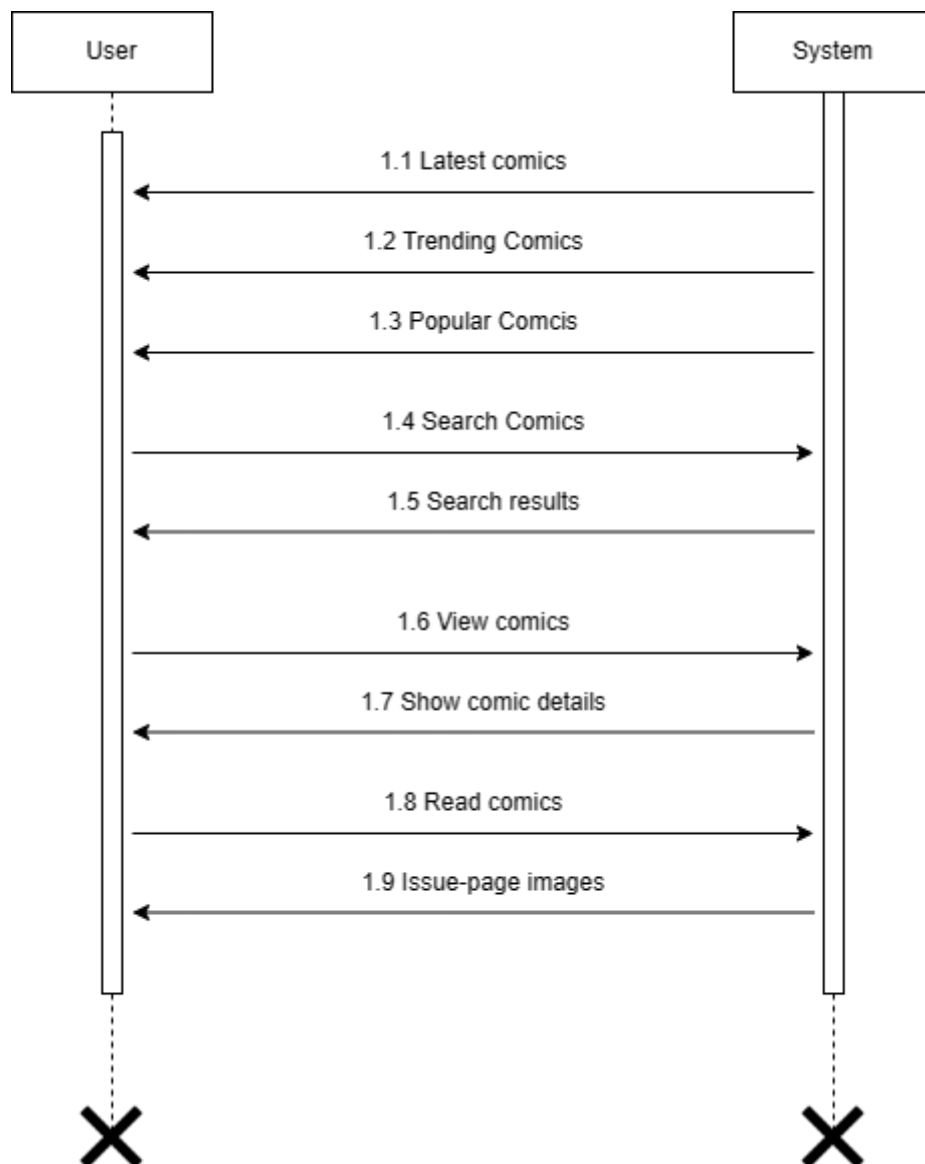
B) E-R Diagram

C) Sequence Diagram

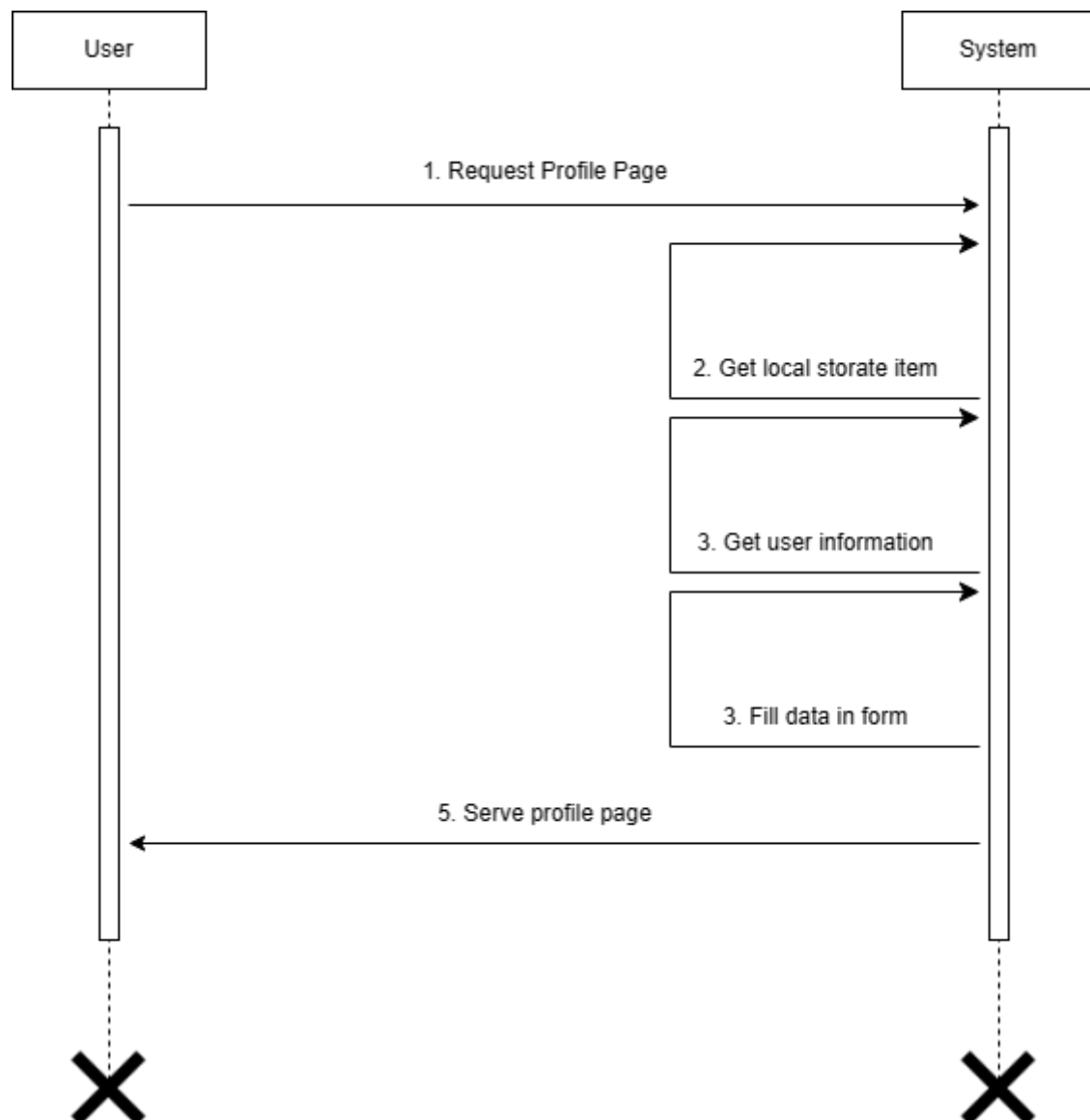
- Sequence diagram for login/signup.

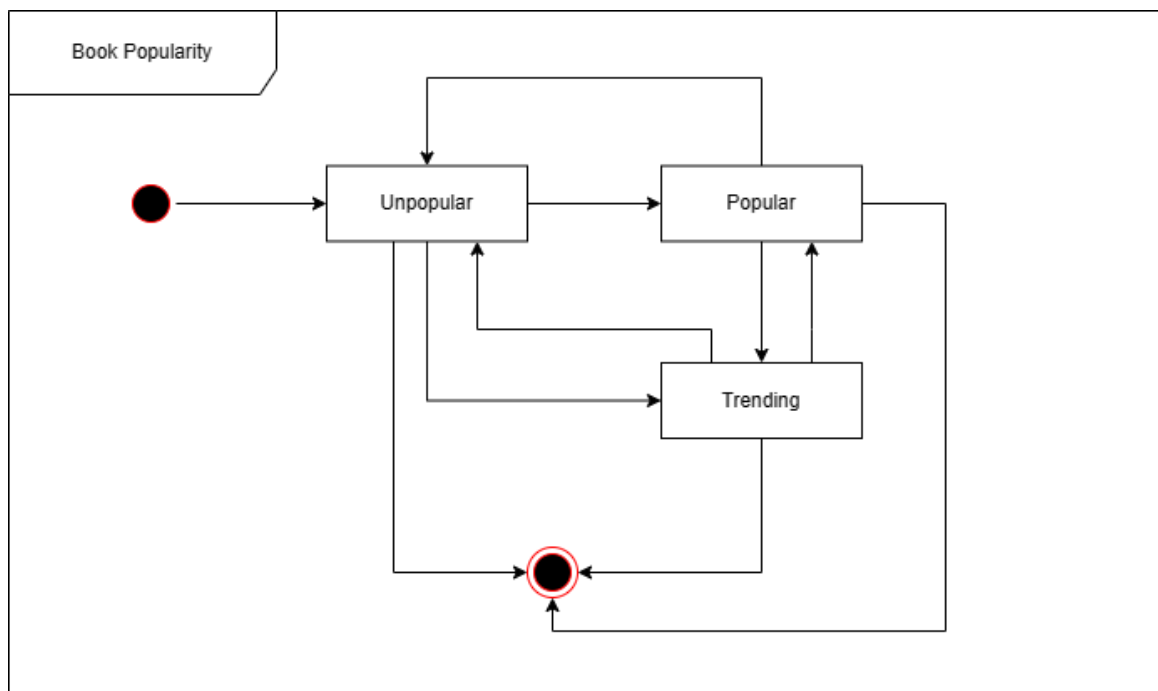
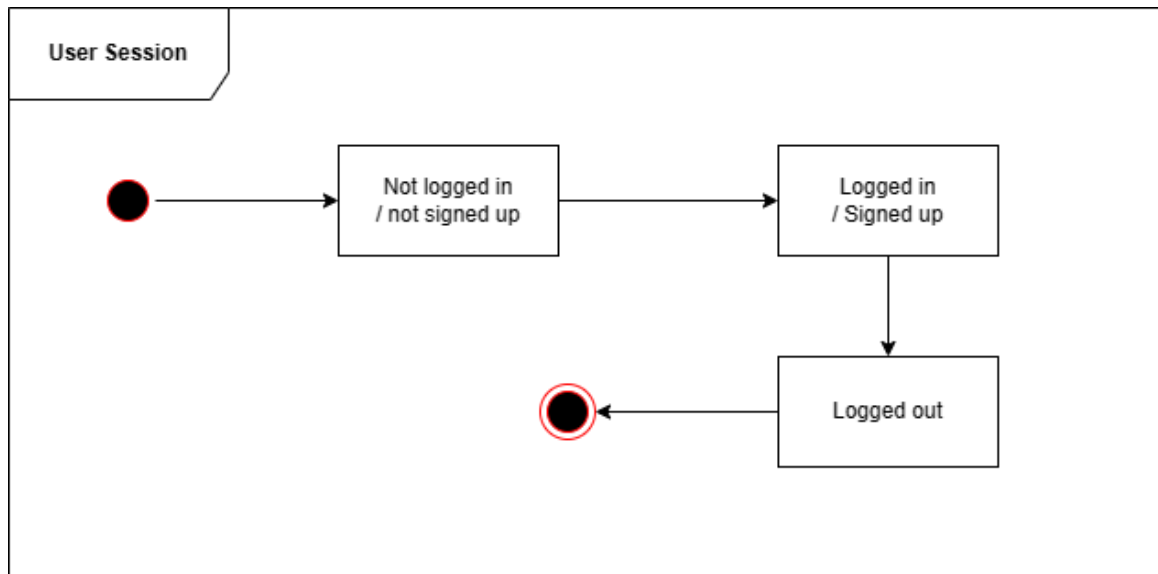


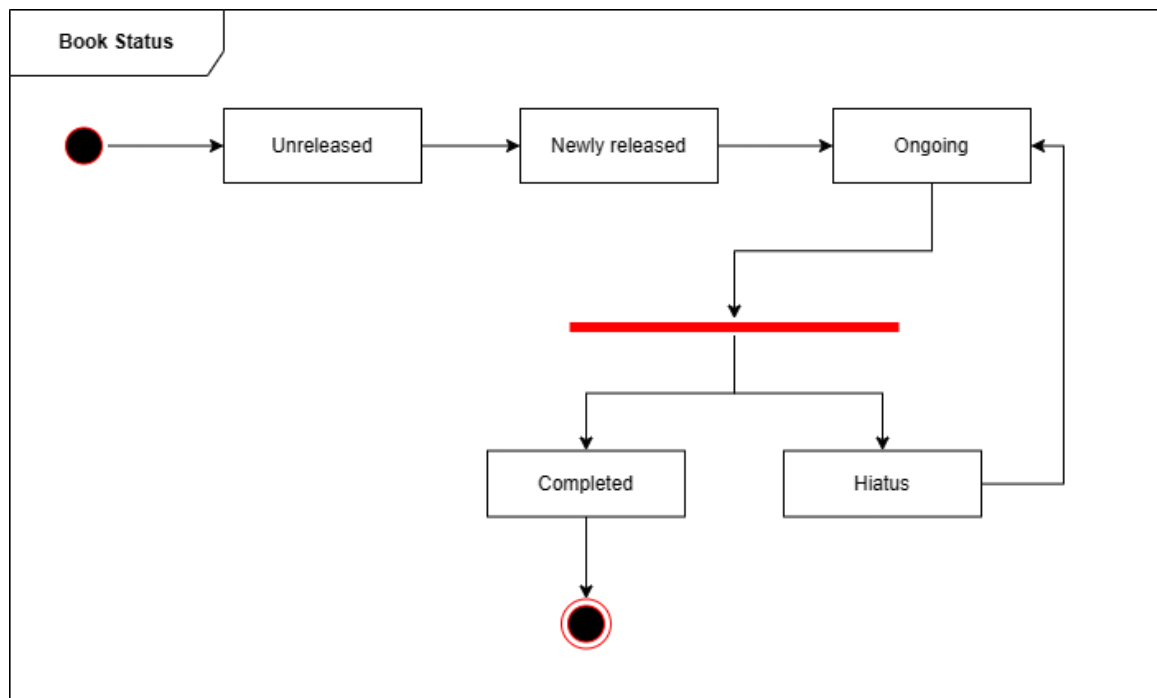
- Sequence diagram for browsing comics.

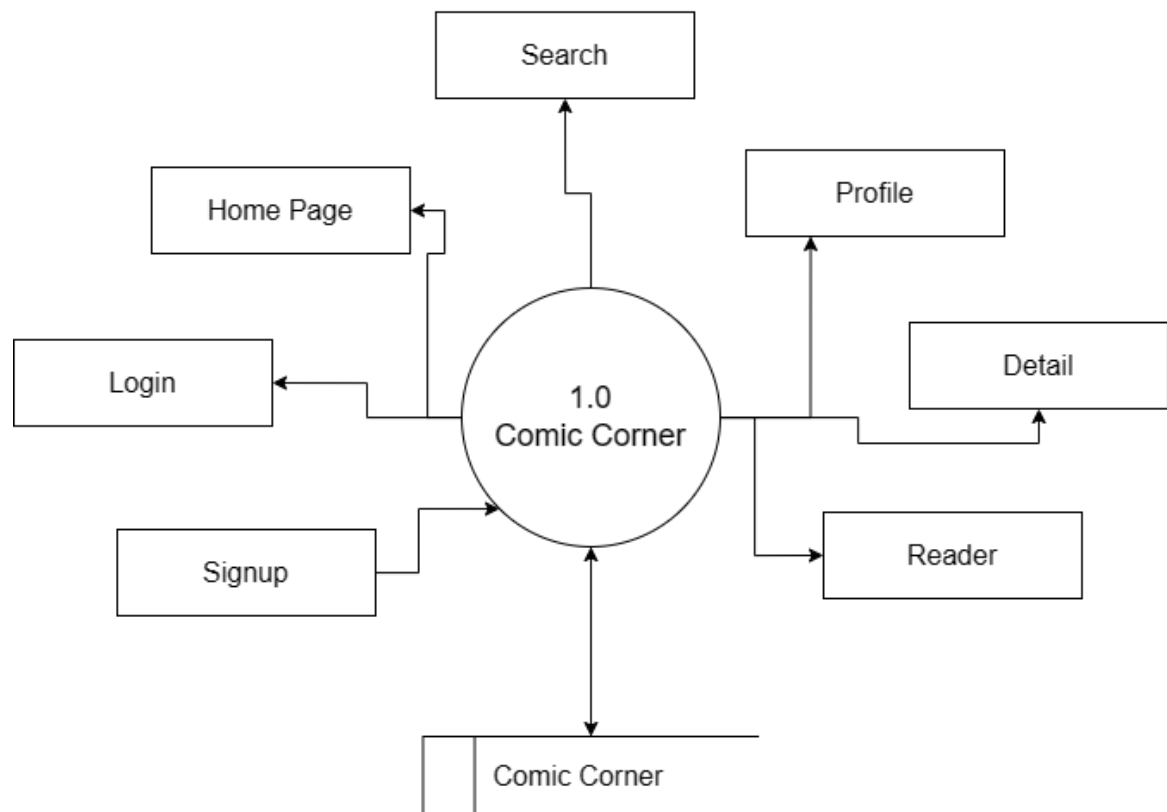


- Sequence diagram for viewing profile

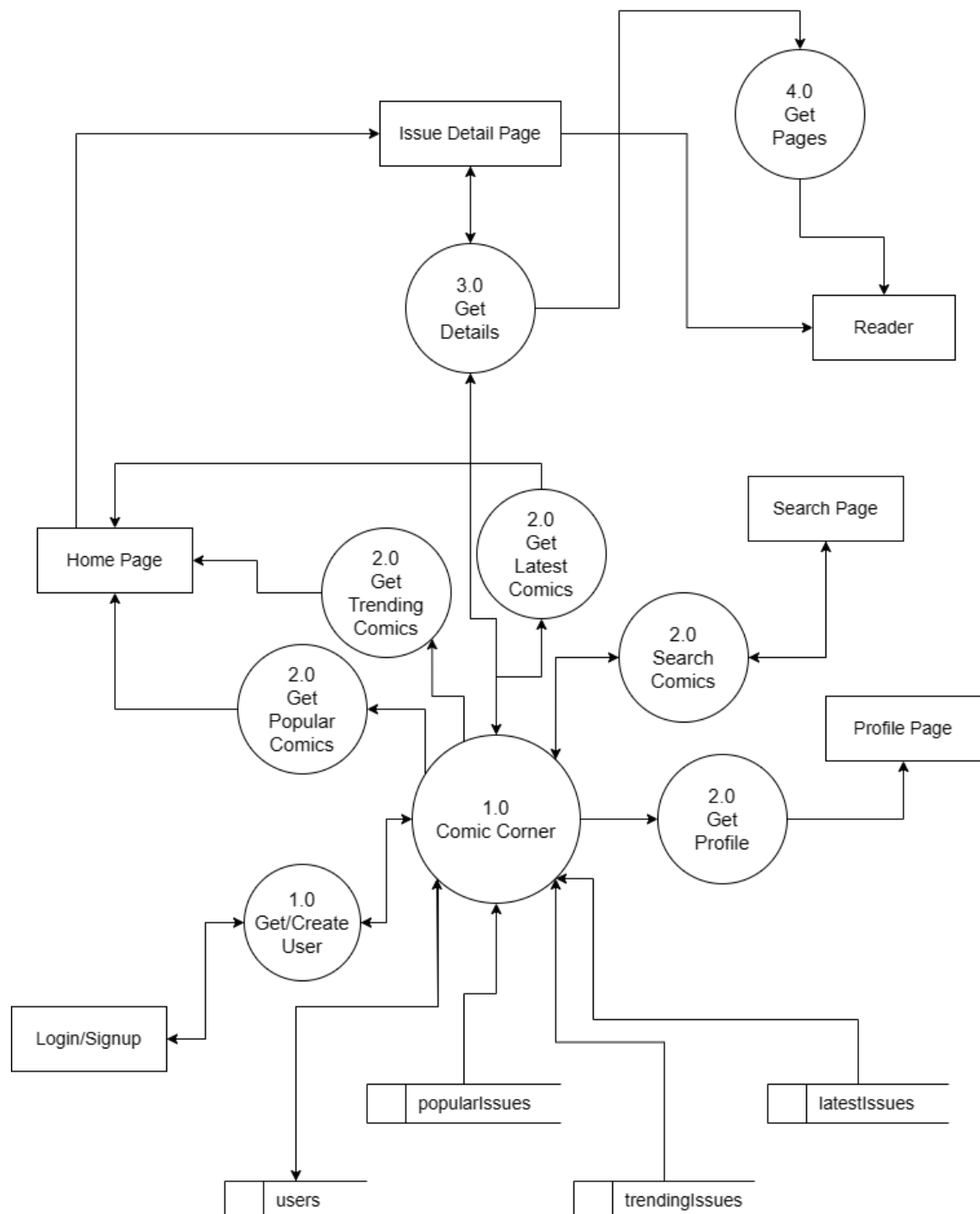


D) State Diagram



E) Dataflow Diagram**DFD Level - 0**

DFD Level - 1



REFERENCES

Websites:

1. <https://stackoverflow.com>
2. <https://react.dev>
3. <https://nodejs.org>
4. <https://mongodb.com>
5. <https://expressjs.com>
6. <https://framermotion.com/motion>
7. <https://chat.openai.com>
8. <https://frontendresource.com>
9. <https://freefrontend.com>
10. <https://npmjs.com>
11. <https://comicvine.gamespot.com>
12. <https://app.diagrams.net>
13. <https://udemy.com>
14. <https://youtube.com>
15. <https://reactrouter.com>
16. <https://fonts.google.com>
17. <https://getbootstrap.com>
18. <https://react-bootstrap.github.io>
19. <https://fontawesome.com>
20. <https://w3schools.com>