

DEERWALK INSTITUTE OF TECHNOLOGY

Tribhuvan University

Institute of Science and Technology



WEB PLATFORM FOR CREATIVE WRITERS

A PROJECT-II REPORT

Submitted to

Department of Computer Science and Information Technology

DWIT College

*In partial fulfillment of the requirements for the Bachelor's Degree in Computer
Science and Information Technology*

Submitted by

Sulav Baskota

20640

11th August, 2022

DWIT College
DEERWALK INSTITUTE OF TECHNOLOGY

SUPERVISOR’S RECOMMENDATION

I hereby recommend that this project prepared under my supervision by SULAV BASKOTA entitled “**WEB PLATFORM FOR CREATIVE WRITERS**” in partial fulfillment of the requirements for the degree of B.Sc. in Computer Science and Information Technology be processed for the evaluation.

.....

Shyam Sunder Khatiwada

Year-In-Charge

DWIT College

Deerwalk Institute of Technology

DWIT College
DEERWALK INSTITUTE OF TECHNOLOGY

STUDENT'S DECLARATION

I hereby declare that I am the only author of this work and that no sources other than that listed here have been used in this work.

.....

Sulav Baskota

11th August, 2022

DWIT College
DEERWALK INSTITUTE OF TECHNOLOGY

LETTER OF APPROVAL

This is to certify that this project prepared by SULAV BASKOTA entitled “**WEB PLATFORM FOR CREATIVE WRITERS**” in partial fulfillment of the requirements for the degree of B.Sc. in Computer Science and Information Technology has been well studied. In our opinion it is satisfactory in the scope and quality as a project for the required degree.

<p>.....</p> <p style="text-align: center;">Shyam Sunder Khatiwada Year-In-Charge DWIT College</p>	<p>.....</p> <p style="text-align: center;">[Examiner] Academic Designation IOST, Tribhuvan University</p>
--	--

ACKNOWLEDGEMENT

I would like to express my sincere thanks to Mr. Shyam Sunder Khatiwada, for his valuable guidance and support in completing my project. His vision, sincerity and motivation have deeply inspired me. He has taught me the methodology to carry out the research and to present the research works as clearly as possible. It was a great privilege and honor to work and study under his guidance. I am extremely grateful for what he has offered me.

Besides my advisor, I would like to thank the rest of my project supervisor committee: Mr. Hitesh Karki, Mr. Ritu Raj Lamsal, Mr. Bijay Babu Regmi, Mr. Saurav Gautam and Mr. Nireesh Dhakal, for their encouragement, insightful comments, and hard questions. Without their support and suggestions, this project would not have been completed.

I would also like to express my gratitude towards our college, Deerwalk Institute of Technology, for giving me this great opportunity to do a project on “Web Platform for Creative Writers”.

Sulav Baskota

Roll No.: 20640

11th August, 2022

ABSTRACT

First time authors have a hard time getting recognition in the competitive space of writing industry with only a few managing to gain fame and recognition. Publishing one's work in this competitive space can be a huge financial burden for novice writers considering that there is no guarantee that their work will succeed. However, with the increase in popularity and consumption of eBooks, amateur authors can take advantage of the internet to share their work with very little financial burden.

The paper also proposes a system for online novel reading, writing and publication that challenges the conventional system and caters to the need of the new internet audience. The chapter-wise method of writing novels suggested in this paper was designed to make it easier for first time authors to enter the book space without having to complete a novel first. This approach which has already been followed by other forms of creative works such as comics, manga, television series has not been effectively introduced in the creative writing industry but has the potential to aid authors gain long term and loyal following of online readers.

This paper analyses the issues an individual author may have if they try to grow their audience on an online platform on their own and proposes a system where multiple authors can share their works for the benefit of both authors and readers.

Keywords: *Web Novel; eBooks; Web Platform; Creative Writing; Node.js;*

TABLE OF CONTENTS

SUPERVISOR’S RECOMMENDATION	ii
STUDENT’S DECLARATION	iii
LETTER OF APPROVAL	iv
ACKNOWLEDGEMENT	v
ABSTRACT.....	vi
LIST OF FIGURES	x
LIST OF TABLES	xi
LIST OF ABBREVIATIONS.....	xii
CHAPTER 1: INTRODUCTION	1
1.1. Overview	1
1.2. Background and Motivation	1
1.3. Problem Statement.....	1
1.4. Objectives	2
1.5. Scope and Limitation	2
1.6. Development Methodology	3
1.7. Outline.....	3
CHAPTER 2: BACKGROUND STUDY AND LITERATURE REVIEW	5
2.1. Background Study.....	5
2.2. Literature Review.....	5
2.3. Current System.....	5
2.4. The problem with Current System.....	6
CHAPTER 3: SYSTEM ANALYSIS.....	7
3.1. Requirement Analysis.....	7

3.1.1. Functional Requirement.....	7
3.1.2. Non-Functional Requirement.....	8
3.2. Feasibility Analysis.....	8
3.2.1. Technical Feasibility	8
3.2.2. Operational Feasibility	8
3.2.3. Economic Feasibility	8
3.2.4. Schedule Feasibility	9
3.3. Analysis.....	10
CHAPTER 4: SYSTEM DESIGN	16
4.1. Design	16
4.1.1. Database Design.....	16
4.1.2. Forms and Report Design	17
4.1.3. Interface and Dialogue Design.....	18
4.2. Algorithm Details.....	18
CHAPTER 5: IMPLEMENTATION AND TESTING	19
5.1. Implementation	19
5.1.1. Tools Used	27
5.1.2. Implementation Details of Modules.....	28
5.2. Testing.....	32
5.2.1. Test Cases for Unit Testing.....	32
5.2.2. Test Cases for System Testing.....	35
5.3. Result Analysis	39
CHAPTER 6: CONCLUSION AND FUTURE RECOMMENDATION.....	40
6.1. Conclusion	40
6.2. Future Recommendation.....	40
REFERENCES	42

APPENDIX.....	43
---------------	----

LIST OF FIGURES

Figure 1: Rapid Application Development Methodology	3
Figure 2: Use Case Diagram	7
Figure 3: Project Gantt Chart	9
Figure 4: ER diagram of the system	10
Figure 5: Level 0 DFD for login and registration	11
Figure 6: Level 0 DFD for creating novel and chapters	12
Figure 7: Level 0 DFD for managing comments	13
Figure 8: Level 0 DFD for managing reviews	14
Figure 9: Level 0 DFD for requesting chapter and managing bookmarks	15
Figure 10: Database Schema Design	17
Figure 11: Separate data fetching for each section of the home page	19
Figure 12: Dictionary object for selecting database search query options	20
Figure 13: Implementation of pagination and sorting.....	21
Figure 14: Using session and user role information to render different components.....	22
Figure 15: Middleware implementation for restricting access to sensitive routes.....	22
Figure 16: Verification of JWT for access control in backend	23
Figure 17: Code for automatically adding or updating bookmark.....	23
Figure 18: Code for accessing font size information from local storage	24
Figure 19: Code for custom credentials providers, JWT and session.....	25
Figure 20: Code for reloading session from client side	26
Figure 21: Sending NextAuth JWT and Session Cookies to backend.....	26
Figure 22: Function to upload image to clouldinary account	27

LIST OF TABLES

Table 1: Test Cases for Unit Testing	32
Table 2: Test Cases for System Testing.....	35

LIST OF ABBREVIATIONS

API Application Programming Interface

DB Database

ER Entity Relationship

JWT JavaScript Web Token

NPM Node Package Manager

SQL Structured Query Language

UI User Interface

UML Unified Modeling Language

URL Uniform Resource Locator

CHAPTER 1: INTRODUCTION

1.1. Overview

An eBook is a book publication made available in digital form and a web novel is a novel published online. With the increase in internet users around the world, the way we consume content has changed drastically in the last couple of decades. The books and novels industry has seen a major shift towards digitization.

The Association of American Publishers reported that, eBook sales revenue was \$345 million from January to October 2010, a 171 percent growth from the same period in 2009, and that eBook sales made up nearly 9 percent of total trade book sales during this period in 2010 [1]. This increase in the number of online readers makes it possible for new authors to introduce their works to potential audience and a web platform where multiple authors can host their works will make it easier to share their audience.

1.2. Background and Motivation

The idea for this project came about with the novel goal of providing up and coming as well as amateur authors a platform to publish their creative works. The writing industry is a competitive one. According to the U.S. Bureau Of Labor Statistics, there were about 49,410 people working as writers and authors in May of 2021 in the U.S., among whom only about 4,040 worked in the Newspaper, Periodical, Book, and Directory Publishers industry [2]. This figure clearly shows how difficult it is to make a living as a writer.

Thus, this project will help up and coming writers to find their audience. It will give them a proper platform where they can test new ideas, and receive feedback from their audience.

1.3. Problem Statement

The book industry is an extremely competitive industry. According to a report from 2004, out of the 1.2 million tracked by “Nielsen Bookscan”, 950,000 titles sold fewer than 99 copies and the average book in America sold about 500 copies [3]. It is difficult for the work of an amateur author to gain any recognition in this competitive market.

According to a survey conducted by UK Authors' Earnings and Contracts in 2018, the top 10 percent of writers still earn about 70% of total earnings in the profession with the average author earning about £16,096 per annum [4]. This amount of earning is not enough for any individual to sustain their livelihood. Only when an author has gained sufficient recognition and reputation in the industry can they hope to make a living out of it. This makes it extremely challenging for novice authors to survive in the industry.

However, even self-published authors can spend up to \$3,000 or more on the book production process [5]. This can be a difficult hurdle for to overcome for many first-time authors. Thus, this project aims to build a free web platform where first-time authors can publish their work.

In addition, in the conventional system of book publication an author needs to finish writing their book before they can approach a publication house. This system is unfavorable to novice writers who may only have a general concept of the story, and prevents them from being able to write a story and share it with an audience in parallel.

1.4. Objectives

The main aspiration of this project are as follows:

- Give amateur authors a free online platform to share their works with readers on the internet and receive feedback from the audience.
- Provide amateur authors with an environment where they can build up their reputation in the industry without having to take on the financial burden associated with publishing hard copy books.
- Provide readers with a single destination where they can read works from up-and-coming authors.
- Allow authors to share their work on a chapter wise basis and enable them to progress the story of their novel as they are sharing it with the readers.

1.5. Scope and Limitation

This system is target to be of help to novice authors. It will provide them with a system to create and publish their writings on the web on a chapter wise basis. It will be of assistance to first-time authors to gain experience and grow by exposing their work to real audience.

1.6. Development Methodology

Rapid Application Development methodology was used for system development. This development methodology consisted of four different phases: requirements planning, user design, construction and cutover. The user design and construction phases are collectively part of an iterative process, where prototype of the system is rapidly developed fulfilling certain system and user requirements.

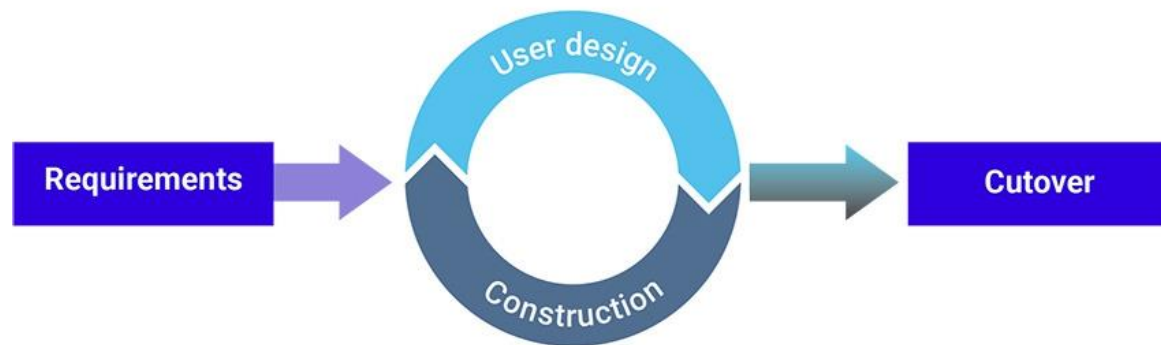


Figure 1: Rapid Application Development Methodology

Kanban methodology is used for project management. This methodology primarily focuses on the work that is currently in progress. A Kanban board is used to visualize the work, which has a three-step workflow: To Do, In Progress, and Done. Kanban cards are used to feature critical information about a particular work item on the Kanban board.

Advantages of Kanban methodology for software development:

- Focus on work that's actively in progress
- Easy to reprioritize work in the backlog
- Simple task management and continuous flow
- Change can happen at any time

1.7. Outline

This project analyses the problem with existing web-based platform for creative writers and develops a system to solve some issues with the existing system. This paper includes the following sections:

Preliminary Section: This section consists of the title page, abstract, table of contents, list of figures, and list of tables.

Introduction: In this section, the overview of the project, the background and motivation of the project, problem statement, its objectives and scope are discussed.

Background Study and Literature Review: This section discusses about the findings in other paper related to this system and discusses in brief the current system and its problem.

System Analysis: This section talks about the functional and non-functional requirements of this project. It also includes diagrams that help to elaborate on the overall design of the system proposed in this project.

System Design: This section deals with the actual designs of the system. On the basis of the designs provided here, the implementation stage can proceed further.

Implementation and Testing: This section is where the actual coding part of the system is described. It also describes some limitation or problems that may have arisen during coding and how it was mitigated.

Conclusion and Future Recommendation: It summarizes the concrete point of the report and provides an insight at the key points of the document. This section also addresses the limitations of the entire system and suggests some workaround or recommendations that could be followed to construct an even efficient system.

References: This section is the end-note of the document and it highlights where certain sources of relevant data and related information were extracted.

CHAPTER 2: BACKGROUND STUDY AND LITERATURE REVIEW

2.1. Background Study

Considering the rising popularity of eBooks in the last decade, this project aims to use existing web technologies to make it easier for novice authors to share their writings among their audience. The project will develop a website to make the author's work available to a wider audience, integrating ability to comment and post reviews for interaction.

2.2. Literature Review

Zakaria Issa Saleh and Ahmad Mashhour's paper on "The Impact of E-Books on the Printed Books: E-books Popularity, Growth and Future", conducted a survey where the finding showed that more than 63% of the authors had published digital books and more than 83% were planning to publish digital books. The paper also states that e-Books will become more popular than the printed books in the future as than half of the participants (60%) in the survey believed that e-book will dominate the market in the future, with only 18% of respondent disagreeing that e-book will be [6]. The system proposed in this paper thus aims to take advantage of this predicted rise in popularity of eBooks and develop a platform for publishing digital novels.

Some online platforms like wattpad.com [7] offer a similar platform to creative writers as discussed in this project, and operate under their own set of rules. These platforms have different rules on how they promote upcoming novels and authors.

2.3. Current System

At present, if an author wants to publish their book traditionally, they will first have to finish writing their book and find an agent who will submit the book to publishing houses. In the event that a publishing house is willing to publish the book, a contract needs to be signed between the publishing house and the author before making revisions and publishing the book.

In addition, most inspiring authors do not have any organized platform where they can test their work, however, some authors do maintain their own blog to promote their works.

2.4. The problem with Current System

It is difficult for inspiring authors to gather interest in their work. Even maintaining a blog to promote their work can only reach so many people. In addition, there is a significant chance of losing existing audience in the case when the author become inactive for a period of time. In addition, novice authors need to finish writing a book before presenting it to a publication house if they want to publish their book in the traditional manner. This is not favorable to first time authors who may not have a complete story and only have some sections of their novel finished but still want to share it with an audience.

CHAPTER 3: SYSTEM ANALYSIS

3.1. Requirement Analysis

In this chapter, the functional and non-functional requirements for the system to be designed are discussed. In addition, this chapter also discusses about the data and process modelling details, and feasibility assessment for the system.

3.1.1. Functional Requirement

The functional requirements of the system are listed below:

- The user shall be able to view and delete bookmarks
- The user shall be able to leave comments on a chapter
- The user shall be able to leave reviews for novels
- The user shall be able to read chapters of a novel
- The author shall be able to create new novels
- The author shall be able to add new chapters
- The author shall be able to edit a novel
- The author shall be able to leave comments on their chapters

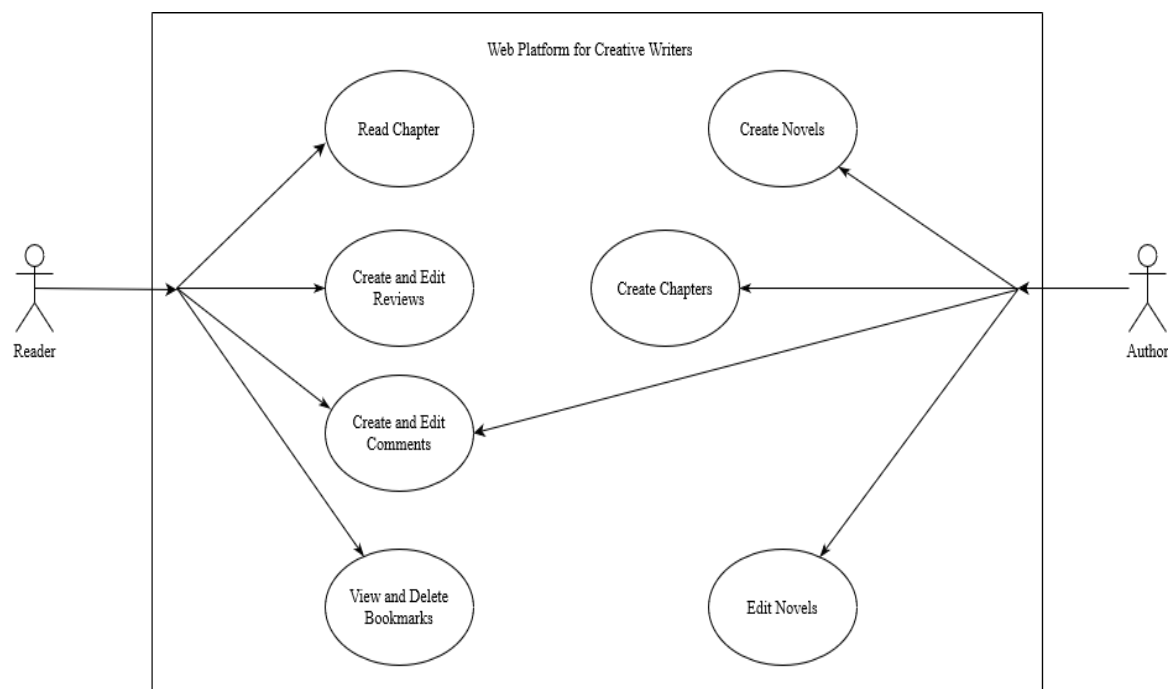


Figure 2: Use Case Diagram

3.1.2. Non-Functional Requirement

The non-functional requirements of the system are listed below:

- The platform must have elegant user interface
- The website must be easy to use
- The connection and service must be reliable

3.2. Feasibility Analysis

3.2.1. Technical Feasibility

The project will be developed using React.js framework with Node.js runtime and Material UI framework for building the frontend of the web platform. A NoSQL database will be used to store information about users, authors, novels, chapters, comments and reviews. The backend will be develop using Next.js.

3.2.2. Operational Feasibility

The system will have a user-friendly interface. Basic knowledge about using web applications on the internet will be enough for both readers and authors to make use of the services provided by the system.

3.2.3. Economic Feasibility

During development and testing the project will be hosted on a local machine and a free tier offer for NoSQL database will be used to minimize cost. In the future, the project will need a subscription from a database provider. Cost will also be incurred for buying a domain and hosting the web application.

For revenue generation, the web application can place advertisements of the web application.

3.2.4. Schedule Feasibility

The time taken to complete the project along with the duration for each task can be observed using the following Gantt chart.

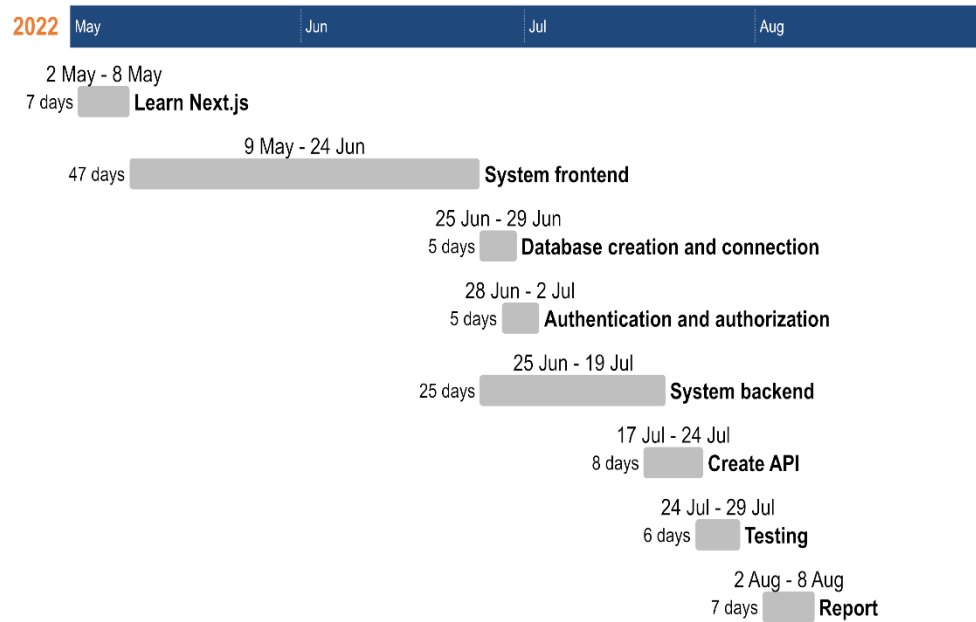


Figure 3: Project Gantt Chart

3.3. Analysis

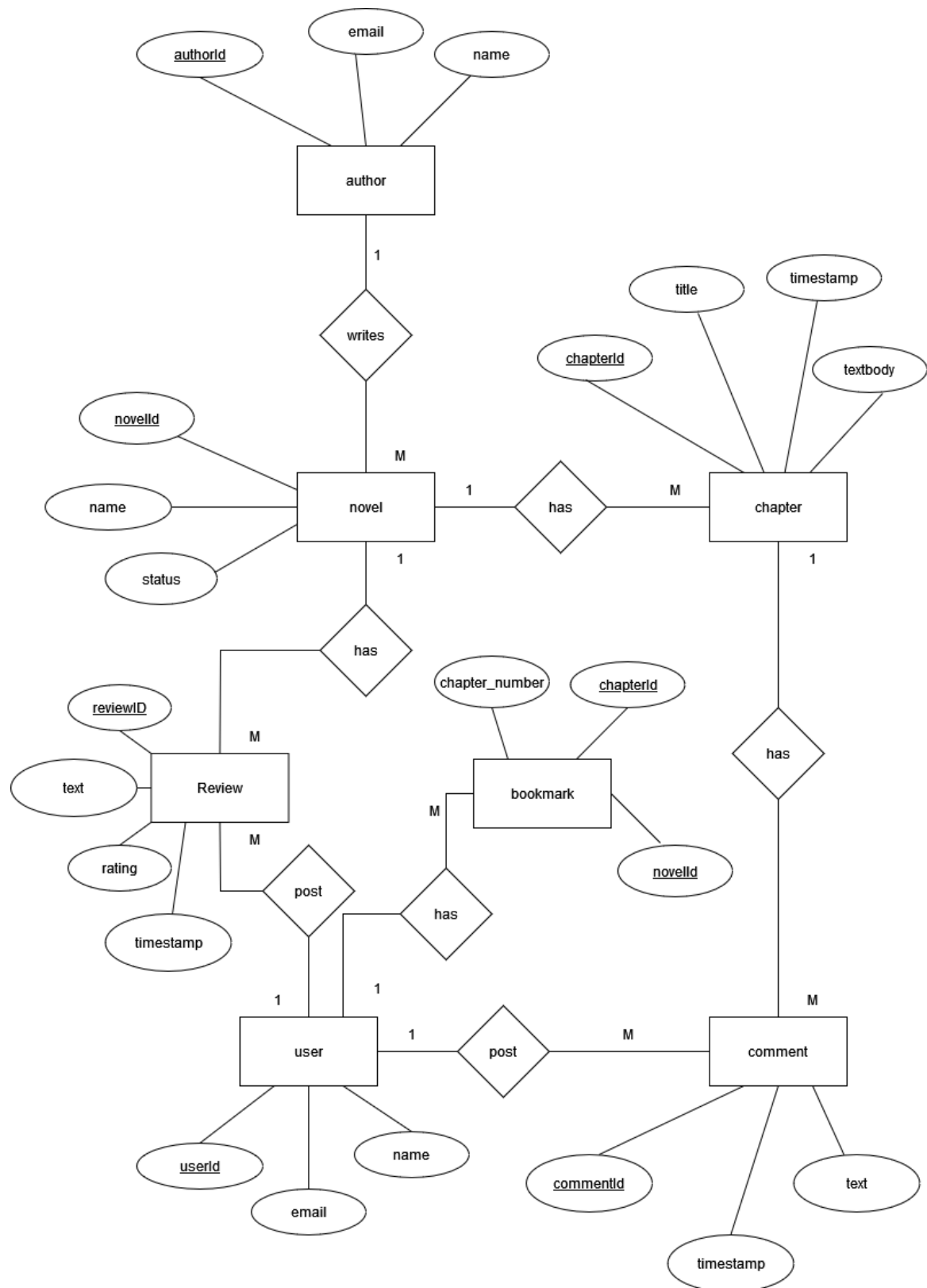


Figure 4: ER diagram of the system

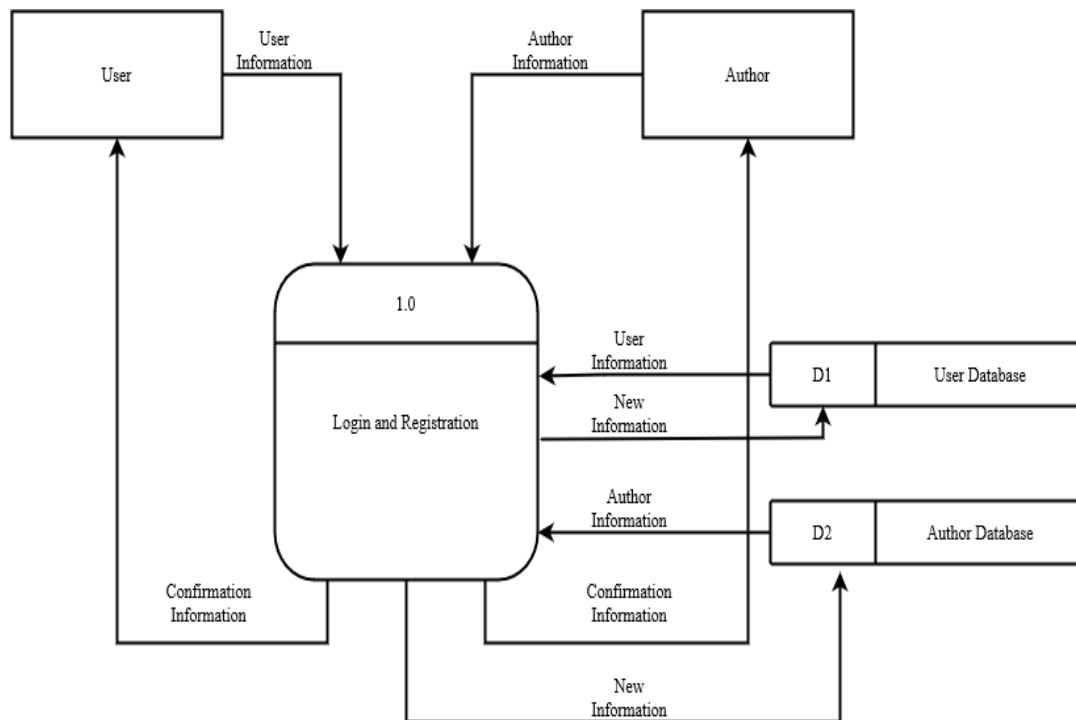


Figure 5: Level 0 DFD for login and registration

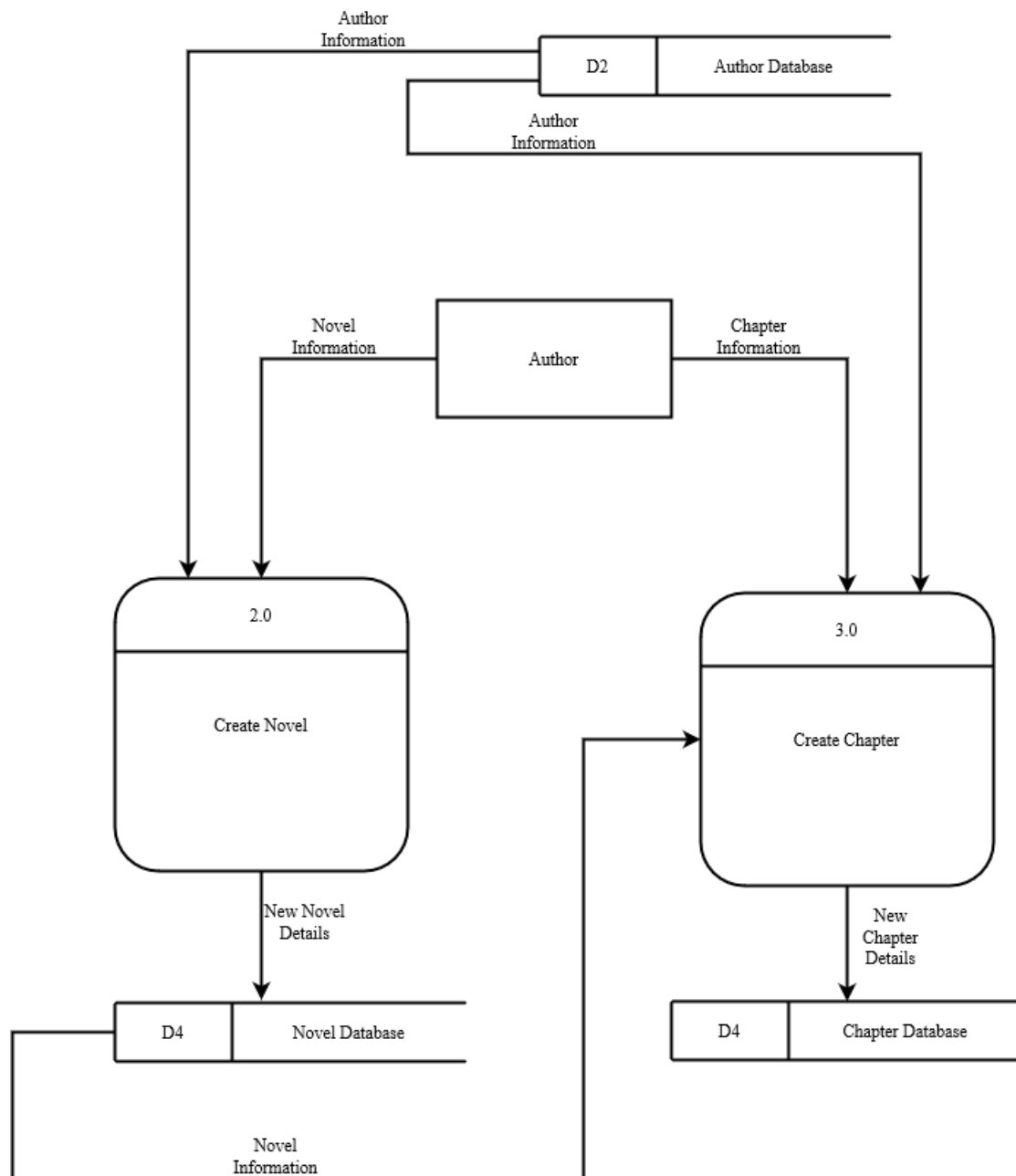


Figure 6: Level 0 DFD for creating novel and chapters

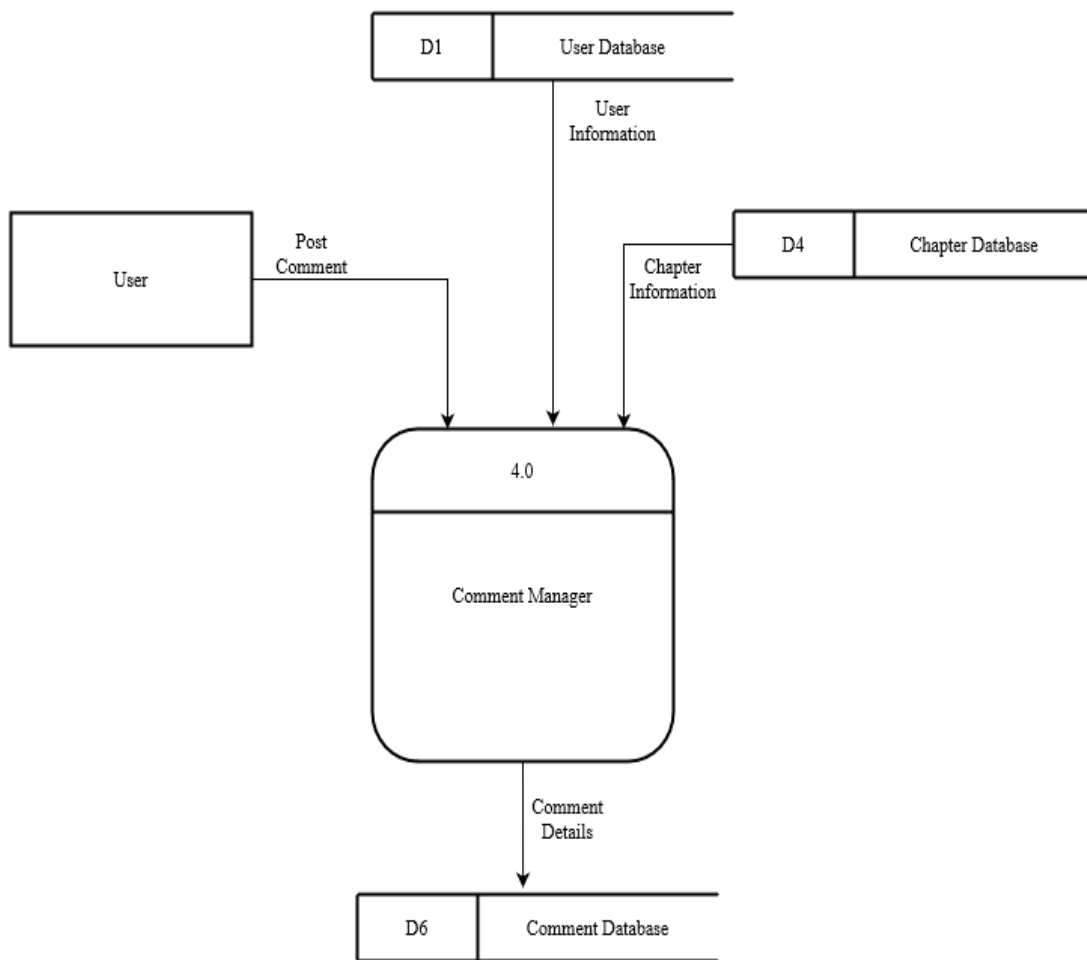


Figure 7: Level 0 DFD for managing comments

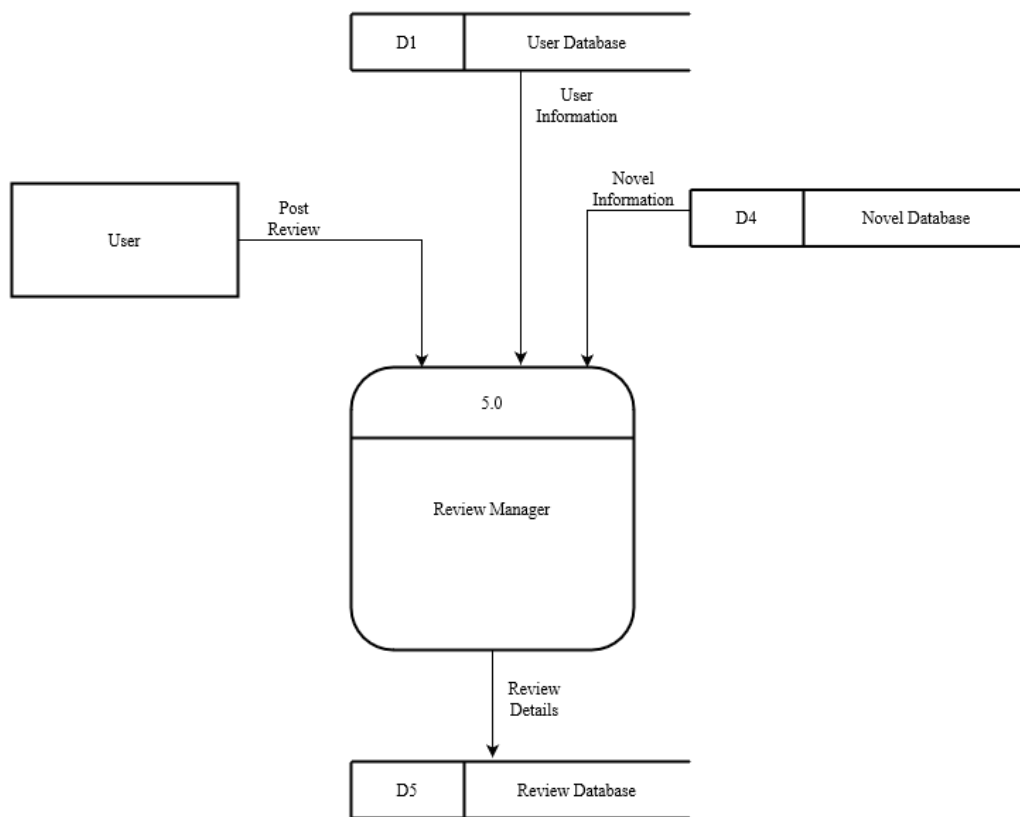


Figure 8: Level 0 DFD for managing reviews

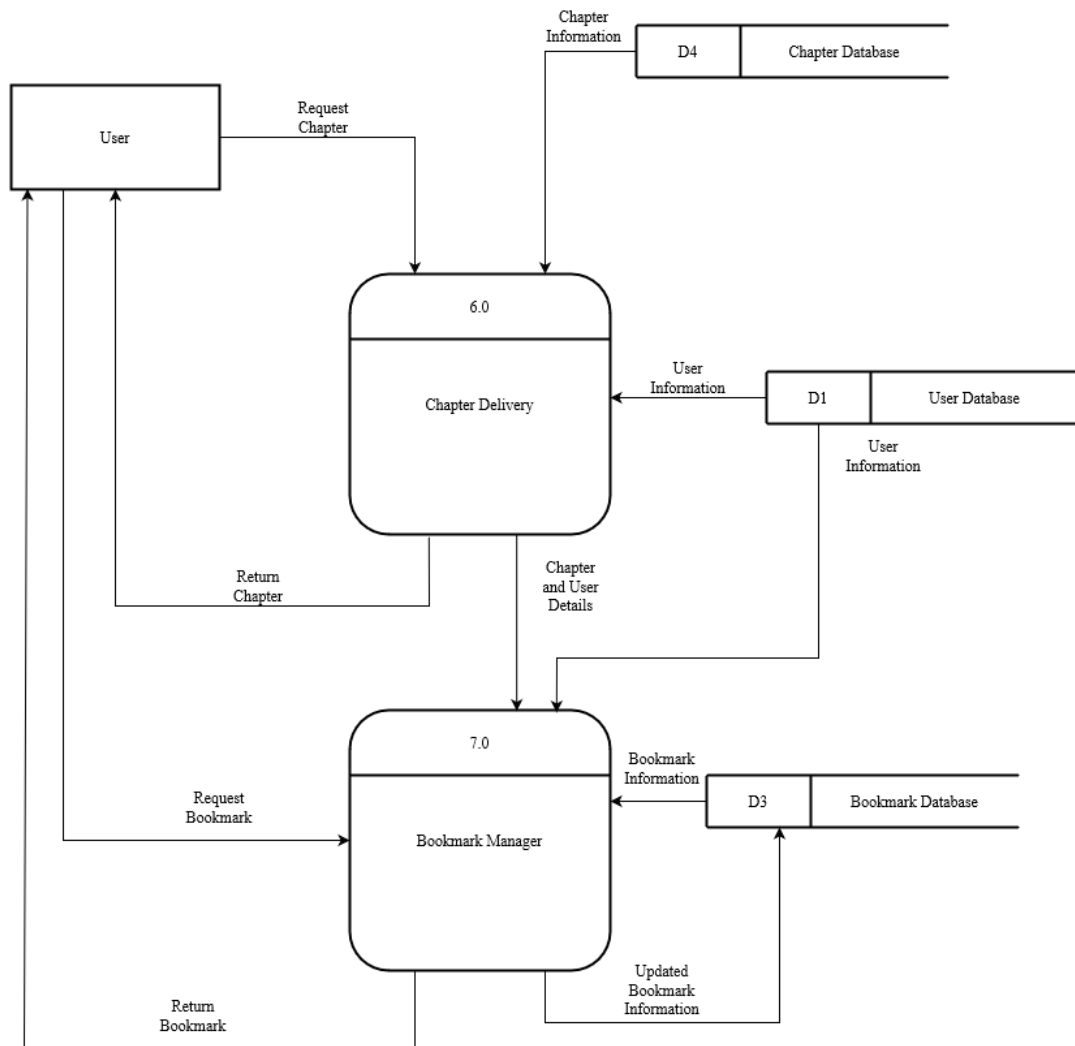


Figure 9: Level 0 DFD for requesting chapter and managing bookmarks

CHAPTER 4: SYSTEM DESIGN

4.1. Design

4.1.1. Database Design

The system has in total six relations, namely: User, Novel, Chapter, Review, Comment, and Bookmark. The relations with their attributes are described below:

- USER (user_id, username, firstname, lastname, email, password, userrole, img_url)
- NOVEL (novel_id, user_id, title, img_url, status, novel_desc, genre, last_chapter, created_on, updated_on, view_count, total_rating, reviews_count)
- CHAPTER (chapter_id, novel_id, title, content, chapter_number, created_on)
- REVIEW (review_id, novel_id, user_id, rating, review, date)
- COMMENT (comment_id, chapter_id, user_id, comment, date)
- BOOKMARK (bookmark_id, user_id, novel_id, bookmark)

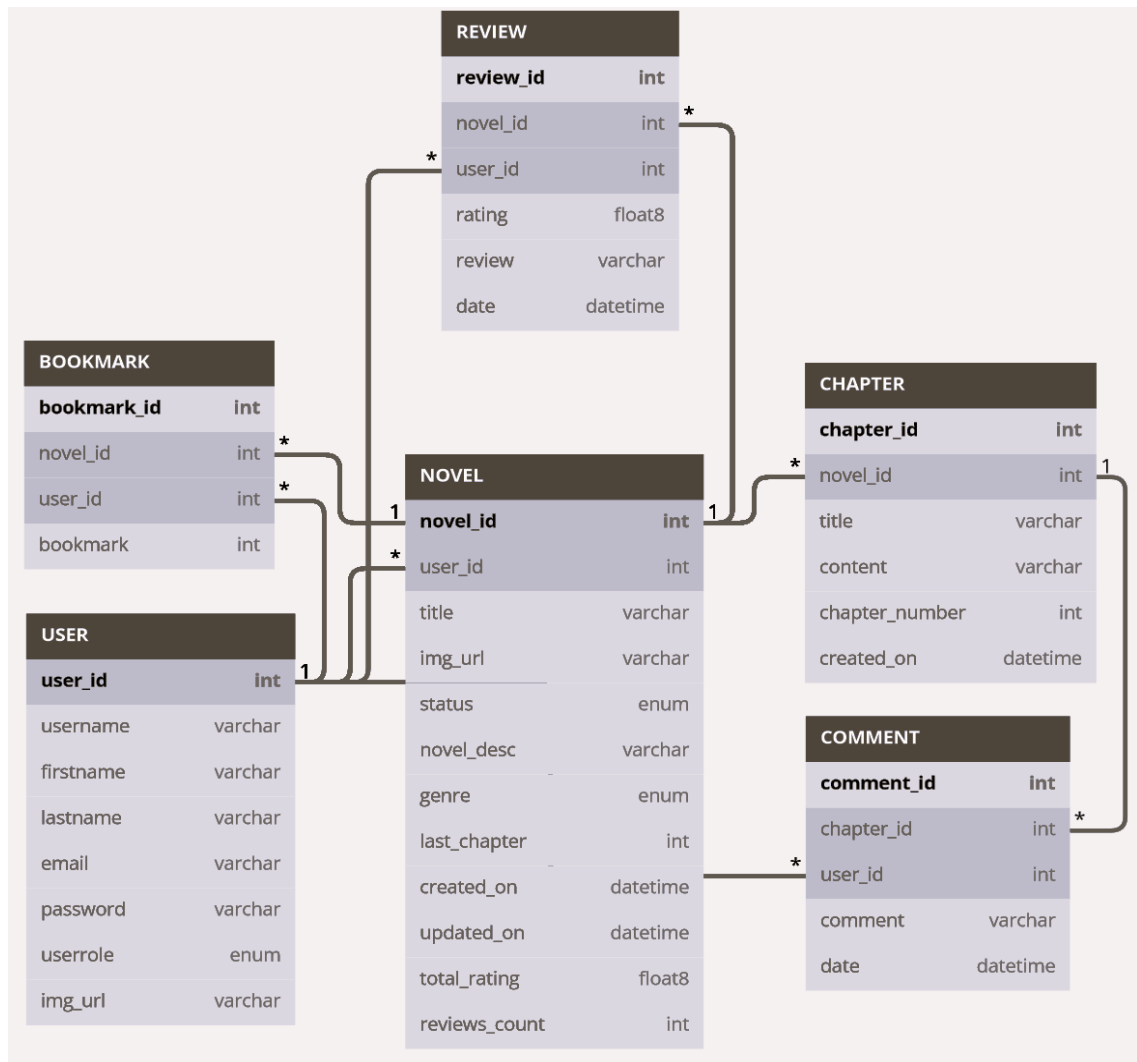


Figure 10: Database Schema Design

4.1.2. Forms and Report Design

The list of forms that have been used in the project are listed below:

- Sign In Form
- Register New User Form
- Forget Password Form
- Change User Profile Details Form
- Create New Novel Form
- Edit Novel Form
- Create Chapter Form

The design and layout of these forms can be viewed in the Appendix section.

4.1.3. Interface and Dialogue Design

Keeping in consideration that the main target audience of the web platform are going to be spending considerable amount of time looking at their screens while using the platform, the entire platform has been designed in dark theme with white text to prevent straining the user's eyes.

In addition, the platform has been designed to be both desktop and mobile friendly. For the desktop version, a traditional navbar has been user, whereas for the mobile version a menu drawer has been implemented. Similarly, different aspect of the web platform such as the carousel for popular novels have different design in the mobile version and the desktop version. The layout of buttons in forms also change depending on whether the user is using a mobile device or a desktop to access the application. Alert and helper text have too been used in forms to guide the users.

The design and layout of the interface and dialogue can be viewed in the Appendix section.

4.2. Algorithm Details

The algorithm to calculate the amount of time since a novel has been updated or a new chapter has been added is given below:

TimeSince(inputDate):

1. Calculate seconds = (presentDate – inputDate) / 1000
2. Calculate interval = (seconds / no of seconds in a year)
3. If interval >=1 return interval + “years ago”
4. Else interval = (seconds / no of seconds in a month)
5. If interval >=1 return interval + “months ago”
6. Else interval = (seconds / no of seconds in a day)
7. If interval >=1 return interval + “days ago”
8. Else interval = (seconds / no of seconds in an hour)
9. If interval >=1 return interval + “hours ago”
10. Else interval = (seconds / no of seconds in a minute)
11. If interval >=1 return interval + “minutes ago”
12. Else return interval + “seconds ago”

CHAPTER 5: IMPLEMENTATION AND TESTING

5.1. Implementation

The system is built using Next.js framework which is a full stack solution for building server-side rendering React.js applications.

The system consists of seven main components: Home, Series, Bookmark, Novel, Chapter, Profile and My Novels. Home, Series, Novel and Chapter components are accessible to all users however the Bookmark component is available to only registered and logged in users, whereas the My Novels components is available to only authors.

The Home page showcases to the readers a list of new novels that have been recently added to the platform that satisfy the requirement for a minimum number of chapters that have been added for the novel. It also displays a carousel showcasing the popular novels in the website according to the total number of views each novel has garnered. Furthermore, the home page has another section displaying novels that have been marked as completed by their respective authors, and a list of up to 20 chapters that have been recently updated.

```
export async function getServerSideProps(context) {
  const hostUrl = process.env.NEXT_PUBLIC_VERCEL_URL;
  const newNovelsRequestUrl = `${hostUrl}/api/novel/get-new-novels`;
  const popularNovelsRequestUrl = `${hostUrl}/api/novel/get-popular-novels`;
  const completedNovelsRequestUrl = `${hostUrl}/api/novel/get-completed-novels`;
  const recentUpdateRequestUrl = `${hostUrl}/api/chapter/get-recent`;

  const fetchHomePageData = async (requestUrl) => {
    const res = await fetch(requestUrl, {
      method: "GET",
      headers: {
        "Content-Type": "application/json",
      },
    }).then((res) => res.json());
    return res.data;
  };

  const newNovels = await fetchHomePageData(newNovelsRequestUrl);
  const popularNovels = await fetchHomePageData(popularNovelsRequestUrl);
  const completedNovels = await fetchHomePageData(completedNovelsRequestUrl);
  const recentlyUpdated = await fetchHomePageData(recentUpdateRequestUrl);

  return {
    props: {
      newNovels: newNovels,
      popularNovels: popularNovels,
      completedNovels: completedNovels,
      recentlyUpdated: recentlyUpdated,
    },
  };
}
```

Figure 11: Separate data fetching for each section of the home page

Keeping in mind future improvements that can be added to the functionality of selecting new, popular and completed novels, the separate API endpoints have been used for each section of the homepage.

To reduce code redundancy an async function has been created to execute the get request that takes the API endpoint URL as parameter and returns the obtained response data after parsing it.

The Series page allows the users to filter the available novels in the platform according to certain filtering criteria. It allows the users to filter the according to novel status and genre and sort them according to either name, popularity, rating or freshness to the platform. These filters are applied collectively, and users have the option to apply only one among the three filters. When filtering by genre instead of using a strict filter, where only those novels that belong to all the selected genres are displayed, a loose filtering is applied to display all novels that have one or more of the selected genres.

At the backend side to reduce the number of if-else statements to search according to the criteria supplied by the users, two dictionary object have been created that contain the search queries as values for different search criteria used as key.

```
import dbConnect from "../../lib/dbConnect";
import Novel from "../../models/Novel";

const statusFilterOption = {
  Any: { $in: ["Ongoing", "Completed", "Hiatus"] },
  Ongoing: "Ongoing",
  Completed: "Completed",
  Hiatus: "Hiatus",
};

const sortByFilterOption = {
  Name: { title: 1 },
  Popular: { view_count: -1 },
  New: { created_on: -1 },
};
```

Figure 12: Dictionary object for selecting database search query options

The Novel page provides users with a detailed description of the novel. It also displays the list of chapters that the author has added for the novel, that the users can read and also list of reviews that other users have left for the novel. The chapters and reviews are displayed in descending order according to the date they were created on and updated on respectively. In addition, pagination has been used to display the list of both reviews and comments.

```
const [page, setPage] = useState(1);
const count = Math.ceil(itemList.length / itemPerPage);

const handleChange = (event, value) => {
  setPage(value);
  setList(itemList.slice((value - 1) * itemPerPage, value * itemPerPage));
};

const handleClick = (event) => {
  setDescending(!descending);
  itemList = itemList.reverse();
  setList(itemList.slice((page - 1) * itemPerPage, page * itemPerPage));
};
```

Figure 13: Implementation of pagination and sorting

A separate sort button for both the list of chapter and reviews has also been implemented so that the users can sort the list of chapters and reviews in ascending or descending order as per their requirement. The sorting function has been implemented by using separate react state to store the sorting order value for chapter list and review list. The review section in this page also provided users with ability to post and edit reviews.

In addition, the novel page also displays different action buttons to users with different roles and verification. For an unlogged in user and users who have not read the novel previously, the novel page provides the user with a Start Reading button that directs them to the first chapter of the novel. For logged in users who have previous read the novel the page provides them with access to a continue reading button that directs them to the last chapter they read for the novel. And for authors a Add New Chapter buttons is provided to allow them to add new chapter to the novel. This feature is made possible by using user session information created at user login to render different components based on user session information and user role information. This page also allows authors to edit their novel description.

```

{!session ? (
  <StartReadingButton novel_id={novel._id} />
) : session.user.id === novel.author_id ? (
  <AddNewChapterButton
    novel_title={novel.title}
    novel_id={novel._id}
    last_chapter={novel.last_chapter}
  />
) : novel.bookmark ? (
  <ContinueReadingButton
    novel_id={novel._id}
    bookmark={novel.bookmark}
  />
) : (
  <StartReadingButton novel_id={novel._id} />
)}

```

Figure 14: Using session and user role information to render different components

The My Novels page is exclusive to users with the role of author. This page displays a list of all novels that the author has published or created on the platform as well as a button to create new novels. Access to this page is restricted using a middleware that check whether or not the user trying to access this page is a logged in or not and whether they have a role of an author or not.

```

import { NextResponse } from "next/server";
import { getToken } from "next-auth/jwt";

export async function middleware(req) {
  const token = await getToken({ req });
  if (req.nextUrl.pathname.startsWith("/user")) {
    if (!token) return NextResponse.redirect(new URL("/auth/signIn", req.url));
  }
  if (req.nextUrl.pathname.startsWith("/author")) {
    if (!token) return NextResponse.redirect(new URL("/auth/signIn", req.url));
    if (token && token.user.userrole !== "author")
      return NextResponse.rewrite(new URL("/401", req.url));
  }
}

```

Figure 15: Middleware implementation for restricting access to sensitive routes

Similarly, the Profile page allows registered users to change their account details such as email address, first name, last name, password and profile picture, The access to this page is also restricted using the middleware to allow access to only logged in users. In addition, JavaScript Web Token is check on the backend side before any changes are committed to the database.

```
import dbConnect from "../../../lib/dbConnect";
import User from "../../../models/User";
import { getToken } from "next-auth/jwt";

export default async function handler(req, res) {
  const token = await getToken({ req });
  if (!token) return res.redirect("/401");
  // if (token) {
  //   // ...
  // }
}
```

Figure 16: Verification of JWT for access control in backend

The Bookmark component is one of the selling features of the system. A bookmark is a record of the last chapter read by the user for a particular novel on the platform. The record is helpful for users to easily pick up reading from where they left the last time. Bookmarks are automatically created at the time when a user visits a chapter from a novel. If a bookmark does not exist for the user for a particular novel, it is then created, otherwise the existing bookmark is updated.

```
if (user_id !== "null") {
  await Bookmark.findOneAndUpdate(
    {
      user: user_id,
      novel: novel_id,
    },
    { bookmark: chapter_number },
    {
      new: true,
      upsert: true,
    }
  );
}
```

Figure 17: Code for automatically adding or updating bookmark

This Bookmark page shows the list of all bookmarks for a user, and also allows then access to a button to continue reading from the last chapter they access for a novel, and another button to delete a bookmark.

The Chapter Component is a dynamic page that is used to display the contents of a chapter to the reader. This page also displays comments made by other users on the chapter which acts as a discussion thread for the chapter. This can be particularly helpful for amateur authors to engage in conversation with their readers and receive valuable feedback from them. The page also contains navigation buttons to navigate to the previous and next chapter in the novel.

This page makes use of a floating speed dial to allow users to increase font size for the text. The information about font size value selected by the user is stored in local storage for use in the future so that, when the user accesses another chapter later one, or some other day, the font size preference of the user is automatically used.

```
useEffect(() => {  
  const fontSize = JSON.parse(localStorage.getItem("fontSize"));  
  if (fontSize) {  
    setFontSize(fontSize);  
  }  
}, []);
```

Figure 18: Code for accessing font size information from local storage

At its core the system is based on the idea of dividing the audience into one of two categories: an author and a reader, with different services and functionality provided to each role.

To achieve this, users were given the option to choose their role during registration. An authentication and authorization system were implemented using NextAuth.js to verify user role and session.

A custom credential provider was created for logging in users with valid credentials, assigning them JWT and establishing session. When logging in users, the role information associated with their account was retrieved from the database along with username and profile image URL, which was then used to create JWT token and session.

NextAuth.js is a serverless authentication system, so it stores the JWT and session information as cookies which can then be accessed both client side and server side for authentication and authorization.

```
export default NextAuth({
  providers: [
    CredentialsProvider({
      name: "Credentials",
      async authorize(credentials, req) {
        await dbConnect();

        const { username, password } = credentials;
        const user = await await User.findOne(
          { username },
          "username password userrole"
        );

        if (!user) return null;
        const isMatch = await bcrypt.compare(password, user.password);
        if (!isMatch) return null;
        const responseData = {
          id: user._id,
          username: user.username,
          userrole: user.userrole,
        };
        return responseData;
      },
    ),
  ],
  callbacks: {
    async jwt({ token, user }) {
      if (user) {
        token.user = user;
      }
      return token;
    },
    async session({ session, token }) {
      if (token) {
        session.user = token.user;
        const res = await User.findById(session.user.id, "imgUrl");
        session.user.imgUrl = res.imgUrl;
      }
      return session;
    },
  },
});
```

Figure 19: Code for custom credentials providers, JWT and session

One of the issues that was resolved in the system, was updating the information about user's image URL saved in the session in the event that the user updated their profile picture. This was achieved by fetching user's image URL from the database in session callback and reloading the session after user updated their profile image information.

```
const reloadSession = () => {  
  const event = new Event("visibilitychange");  
  document.dispatchEvent(event);  
};
```

Figure 20: Code for reloading session from client side

Another issue that was encountered during the development of the system the inability to access the JWT at the backend side of the system. The availability of JWT at the backend side was vital for the implementation of API route protection, especially for requests related to user data. The issue was resolved by sending browser cookies in the GET request inside the "getServerSideProps" function provided by Next.js.

```
export async function getServerSideProps(context) {  
  const session = await getSession(context);  
  const hostUrl = process.env.NEXT_PUBLIC_VERCEL_URL;  
  const id = session.user.id;  
  const requestUrl =  
    hostUrl + "/api/user/account-info/" + encodeURIComponent(id);  
  let accountInfo = {};  
  await fetch(requestUrl, {  
    method: "GET",  
    withCredentials: true,  
    headers: {  
      "Content-Type": "application/json",  
      Cookie: context.req.headers.cookie,  
    },  
  })  
    .then((res) => res.json())  
    .then((res) => (accountInfo = res.data));  
  return {  
    props: { accountInfo: accountInfo },  
  };  
}
```

Figure 21: Sending NextAuth JWT and Session Cookies to backend

In addition to the features mentioned above, the system also provides the registered users who have forgotten their password the feature to change their password. This is implemented by first querying registered users who have requested a password reset information about their account, namely their username and email information. Once these have been verified a random password is generated using a function, and the password is updated in the database. The new password is also emailed to the user's registered email account using "nodemailer" package.

```
const uploadImage = async (imgData) => {  
  const res = await fetch(  
    "https://api.cloudinary.com/v1_1/readhub/image/upload",  
    {  
      method: "POST",  
      body: imgData,  
    }  
  );  
  
  return res;  
};
```

Figure 22: Function to upload image to cloudinary account

Any image uploaded to the website, whether they be user profile picture or book cover image, the images are uploaded to a cloudinary account, and only the image URL is saved in the database.

5.1.1. Tools Used

The following tools and technologies were used to build the web platform for creative writers system:

- Node.js (v 14.19.1): A JavaScript runtime environment. The Node.js runtime environment was used to run the server for the system.
- React.js (v 18.1.0): A JavaScript library for building user interfaces. This library was used to build frontend components and pages.
- @mui/material (v 5.8.0): A package to install Material UI, a popular React UI framework. This package was used to implement styling to the pages built using React.js.

- Next.js (v 12.1.6): A Server-Side Rendering React.js framework. The project was built on top of this framework. It was used to handle frontend routing and also implement API endpoints.
- NextAuth.js (v 4.10.0): A serverless authentication system for Next.js. It was used to validate user login and logout and assign JWT at login. It was also used to create session and implement frontend and API route protection.
- Cloudinary (v 1.30.1): Online Service provider for uploading and fetching images. This service was used to store images uploaded by users such as profile pictures and book cover arts and fetch them whenever necessary.
- mongoose (v 6.4.4): Package for modeling mongodb objects for Node.js. It was used to connect the application to the database and also define the schemas of the different database models.
- mongodb atlas: Cloud hosted mongodb service. This service was used to host the mongodb database on the cloud instead on hosting it locally.
- mongodb: A NoSQL database program. It was used to store data about users, comments, chapters, novels, and reviews.
- bcrypt (v 5.0.1): A password hashing function. This function was used to encrypt the password before saving it in the database and decrypt it during user authentication.
- nodemailer (v 6.7.7): A module for Node.js applications for sending email. This service was used to send an email to users who request a password change.

5.1.2. Implementation Details of Modules

- **Register():**
This function enables the user to create an account for the platform. A user needs to provide information such as their name, username, password, email and role to register. By default, the registration form will select the user role as a reader, which they have to change to author if they want to use the platform to create novels. The username provided by the user is checked for availability in the backend to prevent multiple users from having the same username. In the event that a username is already taken the user is prompted to change their username. The password is hashed using bcrypt before it is stored in the database.

- **SignIn():**

This function takes the user's username and password as input and checks it with the username and password stored in the database. The password in the database is decrypted and matched to the one provided by the user during log in using bcrypt function. If the credentials do not match the user is prompted about the error. If the verification is successful, the user is logged into the website, a JWT token containing information about the user's username and role is created along with a user session that contains additional information about the user's profile picture URL.

- **UpdateProfileDetails():**

The purpose of this function is to allow users to change their account detail information such as their name, email, password and profile picture. When the user sends a request to change the password, they need to provide their old password for validation, and the user is only allowed to change password if the old password is correct. While changing the profile picture the image is first uploaded to the cloudinary folder for storing user profiles and the image URL is stored in the database.

- **CreateNovel():**

This function is exclusive for users with the role of author. It allows authors to create a new novel in the platform. It allows authors to specify a name for the novel, the genres, book cover image and a short novel synopsis.

The availability of the novel name is first checked before the novel is created. If another novel with the proposed name already exists in the platform the user is prompted to pick another name for the novel. When a novel is initially created its status is automatically set to "Ongoing".

- **EditNovel():**

The EditNovel function provides the ability for authors to edit the description of their novel. It allows them to change the novel description, status, cover image but not the novel's name.

- **AddChapter():**

This function allows authors to add a chapter to their novel. The author can specify the chapter's title and content. The chapter number is automatically implemented by using the novel's last chapter as the basis.

- **FilterNovel():**

The FilterNovel function is implemented in the Series page of the application. It allows user to filter the novels on the platform according to criteria such as status, genres and sort the result according to name, popularity, rating and freshness on the platform.

- **DeleteBookmark():**

This function is available only to authenticated user. It is implemented in the Bookmarks page and allows users to delete their bookmarks.

- **PostReview():**

The PostReview function allows users to write review on a novel and also allows them to rate the novel. The value of the rating is used to calculate the average rating for the novel.

- **EditReview():**

This function allows users to change the review they have written for a novel.

- **DeleteReview():**

This function allows the users to delete their reviews.

- **PostComment():**

The PostComment function allows authenticated users to comment on a chapter which serves as a chapter discussion section. User are able to post multiple comments on a single chapter.

- **EditComment():**

This allows users to edit their previous comments.

- **DeleteComment():**

This allows users to delete their comments.

- **ForgotPassword():**

The ForgotPassword function allows users to reset their password in the event that they may have forgotten their password. This function takes the username and email as input as checks whether they are valid or not. If they are valid, it generated a random password and updated the user's password in the database. The updated password is then emailed to the user's email account. The password is emailed using the nodemailer module.

5.2. Testing

5.2.1. Test Cases for Unit Testing

Table 1: Test Cases for Unit Testing

S.N.	Test Case Description	Input	Expected Output	Real Output	Remarks
1	Check user registration with weak password before implementation of password validation.	firstname: Prashant lastname: Neupane username: prashant email: prashant@yahoo.com password: 12345678 confirm password: 12345678 role: reader	Error message stating that password length should be between 8 to 20 characters and contain uppercase and lowercase letters and numbers.	User was successfully able to register.	Fail
2	Check user registration for existing username.	firstname: Oscar lastname: Rossi username: prashant email: readhub99.author@gmail.com password: Prashant12 password: Prashant12 role: author	Username is not available.	Username is not available.	Pass

3	Check user registration with weak password after implementation of password validation.	firstname: Oscar lastname: Rossi username: oscar email: readhub99.author@gmail.com password: 12345678 password: 12345678 role: author	Error message stating that password length should be between 8 to 20 characters and contain uppercase and lowercase letters and numbers.	Error message stating that password length should be between 8 to 20 characters and contain uppercase and lowercase letters and numbers.	Pass
4	Check user registration with strong password after implementation of password validation.	firstname: Oscar lastname: Rossi username: oscar email: readhub99.author@gmail.com password: OscarRossi123 password: OscarRossi123 role: author	Account Successfully Registered.	Account Successfully Registered.	Pass

5	Singin user with incorrect credentials.	username: oscar password: 1234455	Invalid username or password.	Invalid username or password.	Pass
6	Access my novels page without logging in providing parameters in the URL.	http://localhost:3000/author/my-novels	Redirection to http://localhost:3000/auth/signIn	JSON object containing list of novels.	Fail
7	Access my novels page without logging in providing parameters in the URL after implementation of middleware.	http://localhost:3000/author/my-novels	Redirection to http://localhost:3000/auth/signIn	Redirection to http://localhost:3000/auth/signIn	Pass

5.2.2. Test Cases for System Testing

Table 2: Test Cases for System Testing

S.N.	Test Case Description	Input	Expected Output	Real Output	Remarks
1	Check Creating Novel.	Title: City of Sin Genre: [“Fantasy”, “Action”] Description: “Every drop of this family bloodline is stained with sin.”	Novel successfully created and redirection to novel description page.	Novel successfully created and redirection to novel description page.	Pass
2	Editing novel information.	Genre: [“Fantasy”, “Action”, “Mature”] Description: “Every drop of this family bloodline is stained with sin. They are the embodiment of contradiction; calm yet maniacal, with great memories yet often forgetful.”	Update Novel information.	Novel information updated.	Pass

3	Add new chapter to a novel.	<p>Title: “Lu Yun”</p> <p>Novel: 62ef66f7271abd8afff0cab1</p> <p>Content: “Noxious fumes billowed out from the countless cracks that marred the beaten earth, forming a black veil that obscured the red sunset.....”</p>	New Chapter created and listed in chapter list for novel.	New Chapter created and listed in chapter list for novel.	Pass
4	Check automatic addition of bookmark for logged in user accessing a chapter.	Login and read a chapter from any novel.	Bookmark of read chapter shown in the bookmark page.	Bookmark of read chapter shown in the bookmark page.	Pass

5	Review posted by user should be reflected in the novel's rating.	<p>Rating: 3.5</p> <p>Novel: 62ef7815c73b251 7fd58eabc</p> <p>Review: "I have enjoyed this author's previous works and from what I have read ..."</p>	Change in Novel rating.	Novel rating changed.	Pass
6	Filter novels according to filter criteria.	<p>Status: "Completed"</p> <p>Sort By: "Popular"</p> <p>Genres: ["Fantasy"]</p>	Fetch list of all novels that fit the search criteria.	List of all novels that fitted the search criteria.	Pass
7	Updating user profile picture changes the profile picture URL in session.	Upload a new profile picture.	Image URL in session updated to new image URL.	Image URL in session not updated.	Fail

8	Updating user profile picture changes the profile picture URL in session after implementing session callback for fetching profile picture URL and reloading session after update.	Upload a new profile picture.	Image URL in session updated to new image URL.	Image URL in session updated.	Pass
---	---	-------------------------------	--	-------------------------------	------

5.3. Result Analysis

To test the system 5 different accounts with the role of author and 5 different accounts with the role of reader were created.

The accounts having the role of author were successfully able to create new novels. They were able to access list of all novels that they have created in the My Novels page, and were successfully able edit the novel details for their novel. The authors were also able to add chapters to their novels, and any chapter added was successfully displayed in the recently updated section of the homepage. Just as the system intended, the authors were able to leave comments on chapters belonging to their and other's novel, but were denied from leaving reviews for their own novels.

The accounts with the role of reader were used to access and read chapters of a novel. It was observed that whenever a user accessed a chapter, a bookmark was automatically created for the novel and the view count for the novel was also successfully increased. These accounts were also used to leave comments on different chapter and review different novels. The ratings provided by these users were successfully reflected in the total rating for the corresponding novels.

In addition to testing reader and author functionality, the system was also successfully able to implement authentication and authorization for restricted pages. The accounts with role of readers were unable to access the services for creating and editing novel and adding chapter, whereas unregistered users were unable to access the profile page and bookmarks page in addition to the pages restricted to authors.

Thus, the system was able to deliver upon its intended features, functions and services.

CHAPTER 6: CONCLUSION AND FUTURE RECOMMENDATION

6.1. Conclusion

The system, “Web Platform for Creative Writers”, proposed in this paper aims to aid novice writers enter the competitive book industry. The intention of this paper was to build a system that allowed new writers to engage with their audience to gain experience and feedback.

It embraces the trend of book publication moving away from the traditional paper-based distribution to online digital distribution. Furthermore, this paper also caters to the idea that novels do not have to be finished before they are published for the audience to read, but instead can follow the approach used by other creative works such as comics and TV series, and be updated on a regular basis, allowing amateur authors more control over the progression of their stories.

The approach proposed in this paper for writing novels on the web has the potential to develop into a robust system of its own, and may change the way we read digital novels in the future.

6.2. Future Recommendation

The system proposed in this paper provides amateur authors to expose their ideas to potential audience, however it does not provide a system for authors to gain monetary gain for publishing their work on the platform. To provide even more benefit and incentive for authors to use and publish on the platform a sub-system can be implemented that will allow authors to restrict access to their content in exchange for some in platform currency like in a game.

Readers should be able to buy virtual tokens using real currency and use them to access restricted content. To allow users who are unable to pay for accessing the content, users can then be rewarded fixed points for daily log in which they can then use for accessing restricted content. The authors can be paid a cut of the revenue generated by advertisement

and by selling virtual tokens based on the points spent by users for accessing content of a particular author.

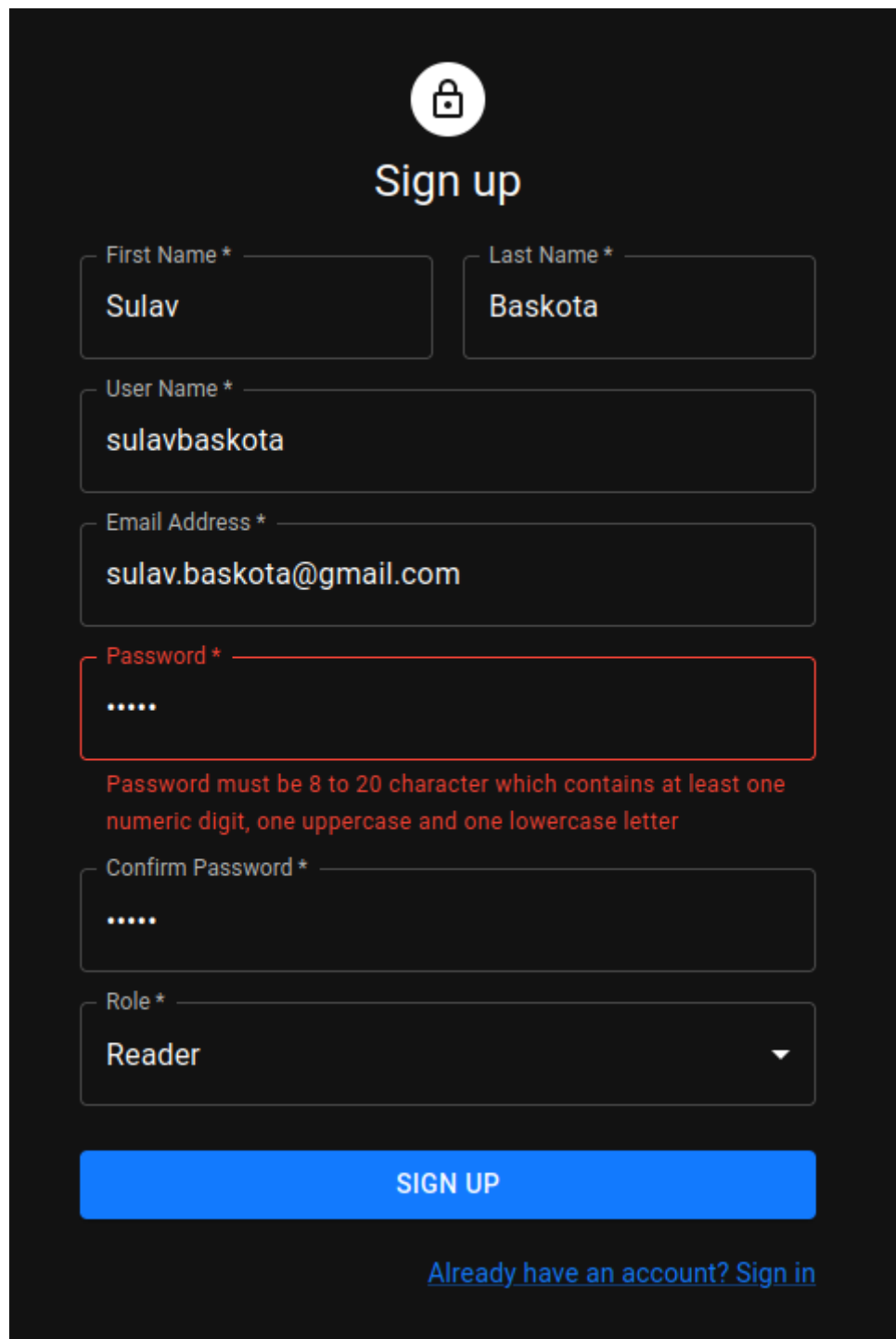
A system for keeping track of all restricted content accessed by the user can be added to prevent users from having to spend money to access the same content again.

In addition, options for users to donate to their favorite authors can also be added in the system.


REFERENCES

- [1] Hailiang Chen, Yu Jeffrey Hu, Michael D. Smith, "The Impact of Ebook Distribution on Print Sales: Analysis of a Natural Experiment," August 2017.
- [2] "Occupational Employment and Wage Statistics," United States Department of Labor, 31 3 2022. [Online]. Available: [https://www.bls.gov/oes/current/oes273043.htm#\(1\)](https://www.bls.gov/oes/current/oes273043.htm#(1)). [Accessed 27 4 2022].
- [3] C. Anderson, "A Bookselling Tail," 14 7 2006. [Online]. Available: <https://www.publishersweekly.com/pw/by-topic/columns-and-blogs/soapbox/article/6153-a-bookselling-tail.html>. [Accessed 24 4 2022].
- [4] Martin Kretschmer, Andres Azqueta Gavaldon, Jaakko Miettinen, Sukhpreet Singh, "UK Authors' Earnings and Contracts 2018: A Survey of 50,000 Writers," 7 June 2019.
- [5] B. R. Pope, "Self-Publishing vs Traditional Publishing: How to Choose," Self-Publishing SCHOOL, [Online]. Available: <https://self-publishingschool.com/self-publishing-vs-traditional-publishing/>. [Accessed 2022 04 27].
- [6] Zakaria Issa Saleh, Ahmad Shaher Mashhour, "The Impact Of E-Books On The Printed Books: E-Books Popularity, Growth," 18 Oct 2015.
- [7] "Home," Wattpad, [Online]. Available: <https://www.wattpad.com/home>. [Accessed 11 6 2022].

APPENDIX



A sign-up form on a dark background. At the top is a white circular icon with a padlock. Below it is the text "Sign up". The form contains several input fields: "First Name *" with the value "Sulav", "Last Name *" with the value "Baskota", "User Name *" with the value "sulavbaskota", "Email Address *" with the value "sulav.baskota@gmail.com", "Password *" with five dots and a red error message below it, "Confirm Password *" with five dots, and a "Role *" dropdown menu with "Reader" selected. A blue "SIGN UP" button is at the bottom. A link "Already have an account? Sign in" is at the bottom right.



Sign up

First Name *

Last Name *

User Name *

Email Address *

Password *

Password must be 8 to 20 character which contains at least one numeric digit, one uppercase and one lowercase letter

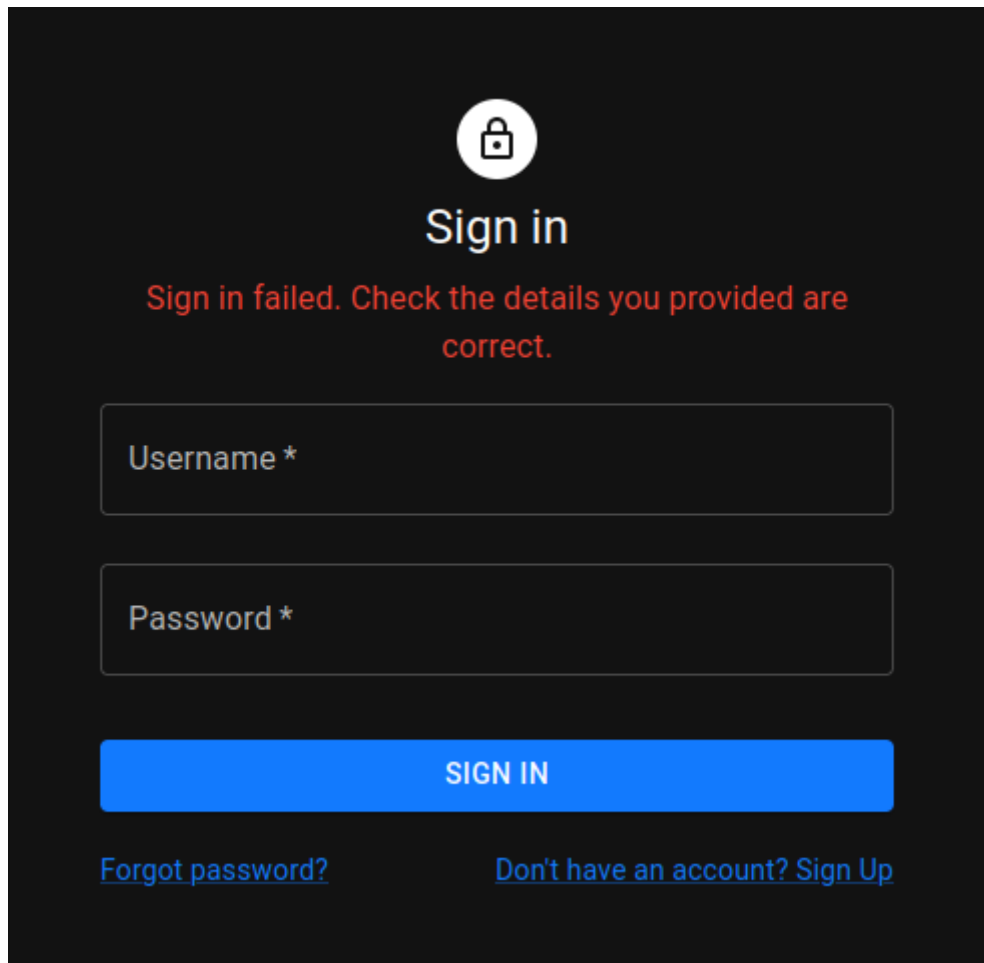
Confirm Password *

Role *


[SIGN UP](#)

[Already have an account? Sign in](#)

Figure A: Sign up form with error message



The image shows a dark-themed sign-in form. At the top center is a white circular icon containing a padlock. Below the icon is the text "Sign in" in a large, white, sans-serif font. Underneath this is an error message in a smaller, red, sans-serif font: "Sign in failed. Check the details you provided are correct." Below the error message are two input fields. The first field is labeled "Username *" and the second is labeled "Password *", both in a light gray, sans-serif font. Below the input fields is a bright blue rectangular button with the text "SIGN IN" in white, uppercase, sans-serif font. At the bottom of the form are two links in a light blue, sans-serif font: "Forgot password?" on the left and "Don't have an account? Sign Up" on the right.



Sign in

Sign in failed. Check the details you provided are correct.

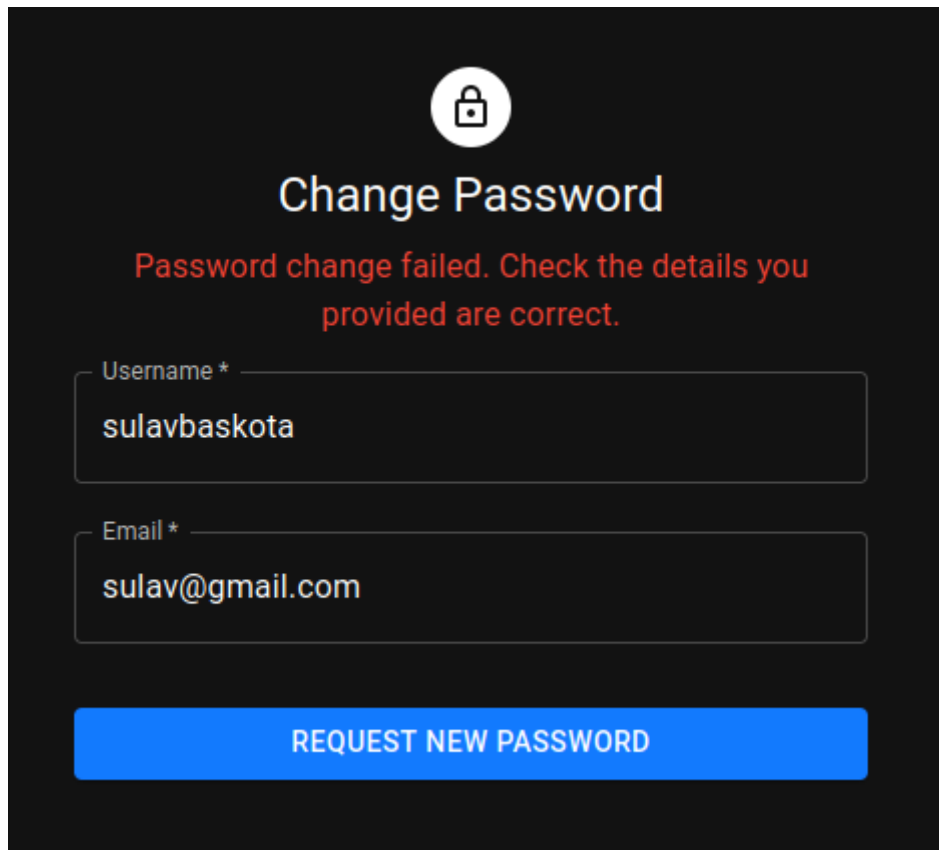
Username *

Password *


SIGN IN

[Forgot password?](#) [Don't have an account? Sign Up](#)

Figure B: Sing in form with error message



The image shows a 'Change Password' form on a dark background. At the top is a white circular icon with a padlock. Below it is the title 'Change Password' in white. A red error message reads: 'Password change failed. Check the details you provided are correct.' There are two input fields: 'Username *' containing 'sulavbaskota' and 'Email *' containing 'sulav@gmail.com'. At the bottom is a blue button with the text 'REQUEST NEW PASSWORD'.



Change Password


Password change failed. Check the details you provided are correct.

Username *

Email *

[REQUEST NEW PASSWORD](#)

Figure C: Change Password form with error message

 **SULAVBASKOTA**

Change Name

Firstname *

Sulav

Lastname *

Baskota

CHANGE NAME

Change Email

Email *

sulav.baskota@gmail.com

CHANGE EMAIL ADDRESS

Change Password

Old Password *


New Password *

Confirm Password *

CHANGE PASSWORD

Upload Profile Picture

No file selected.

 **UPLOAD**

CHANGE PROFILE PICTURE

CANCEL

Figure D: Form for updating user account information

Novel Details

Novel Name *

Deadman

Novel Description *

Even after the bombs fell, even after countries unleashed weapons beyond imagining on one another, even after lawlessness became the norm, people still expect their mail.

As a ~~deadman~~, I'm uniquely suited to deliver it. Rads don't bother me, and people who try to hurt me typically wind up dead. Sure, humans may not like dealing with a face like mine, but hey, you can't beat my express rates.

Select Genres

Thriller

Dystopian

Sci Fi


Add Genre

Upload Book Cover Art

51018-deadman-a-post-apocalyptic-litrpg.jpg

UPLOAD

Image Preview:



CANCEL

CREATE NOVEL

Figure E: Create novel form

47

Novel Details

Novel Name *

The Perfect Run

Novel Description *

Ryan "Quicksave" Romano is an eccentric adventurer with a strange power: he can create a save-point in time and redo his life whenever he dies. Arriving in New Rome, the glitzy capital of sin of a rebuilding Europe, he finds the city torn between mega-corporations, sponsored heroes, superpowered criminals, and true monsters. It's a time of chaos, where potions can grant the power to rule the world and dangers lurk everywhere.

Ryan only sees different routes; and from Hero to Villain, he has to try them all. Only then will he achieve his perfect ending... no matter how many loops it takes.

Status

☒ Ongoing ☐ Completed ☐ Hiatus

Select Genres

Adventure

Comedy

Mystery

Add Genre

Upload New Book Cover Art

No file selected.

UPLOAD

CANCEL

SAVE CHANGES

Figure F: Update novel form

Chapter Details

Novel Title

The Perfect Run

Chapter Number

9

Chapter Title *

Home

Chapter Content *

I took the kettle off the burner and poured it into the mug in front of me. I had to compensate a little as I poured for the gentle bounce of my boat on the water. I took out a pack of military ration instacaf and poured it into the steaming water, then stirred it gently with a small spoon I kept on the counter for just that purpose. I lifted the mug and took it out onto the deck, enjoying a nice long sip as I stepped into the sunlight.

It was a clear day so far, aside from the green crackling mist that drifted gently above the water. I listened for a while to the sounds of frogs and crickets greeting the day, and just stood and enjoyed the warmth seeping into me from the sun as well as the radiation that surrounded me.

I lived in what was basically my own personal deadzone. Between postman runs and courier missions I used to explore and map the different zones, marking what made them dangerous aside from the high levels of rads, and passing that information on to guides and Undertakers. When I'd first found this one, a medium sized lake in the middle of a forest, I'd found a decently sized houseboat still floating on the water and decided to make it my own. It had taken some time to make it work. I'd started with a leaky old nuclear gen that a settlement was more than willing to let me have, some lumber, and some wax to seal and patch the surprisingly minor damage it had sustained. After that I started filling it with

CANCEL

ADD CHAPTER

Figure G: Add new chapter form

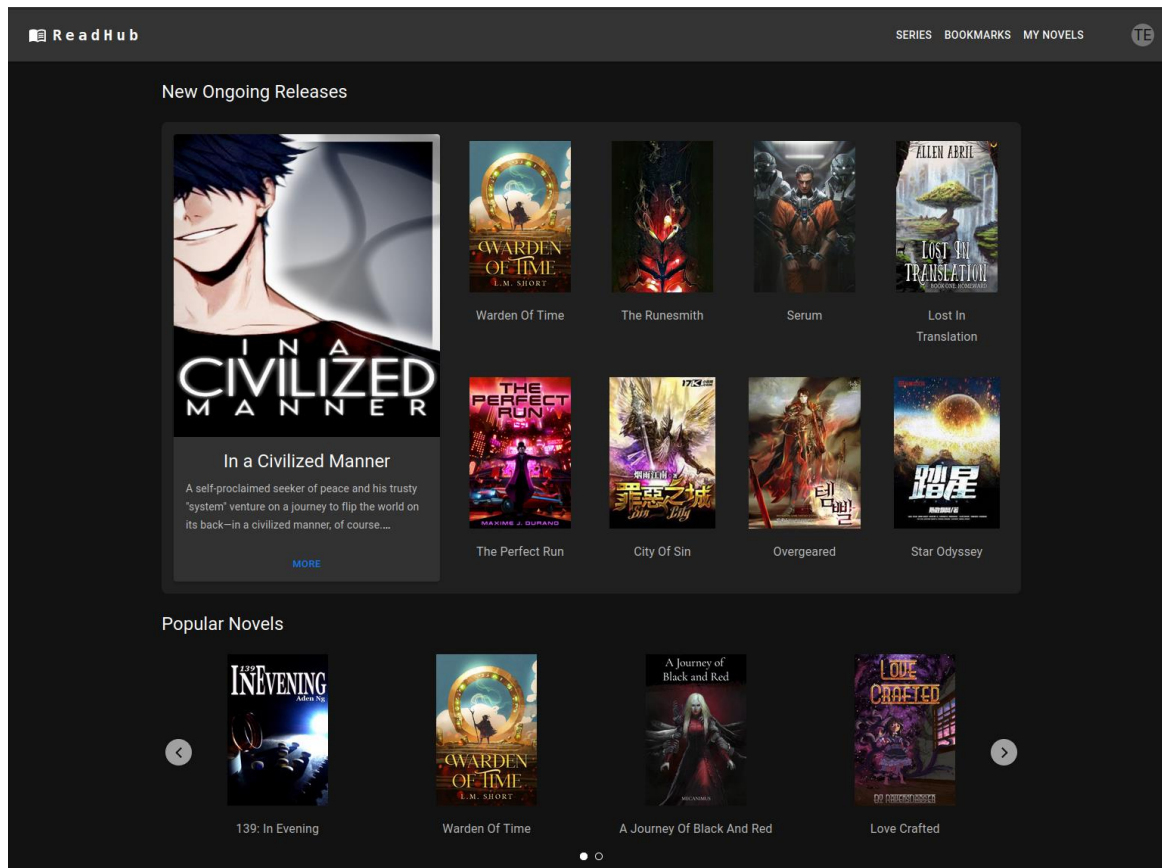



Figure H: Desktop view of home page



Ongoing

In a Civilized Manner

4 Chapters

Author: Chares Akamu

Mostly Positive ★★★★★ ☆ [3.8 stars](#)

A self-proclaimed seeker of peace and his trusty "system" venture on a journey to flip the world on its back—in a civilized manner, of course.

Edris has always aimed for a peaceful, carefree life, but the world around him? Not so much. It started with encountering the amnesic, system-like entity in his head. From there, only absurdities followed. Aristocratic conspiracies, mismatched identities, intercontinental wars... Hold on—did he just get called a prophet? In any case, as the world descends into chaos, Edris's carefree life may be just a bit further away than expected.

[Less](#)

Western

True Crime

Mature

START READING

CHAPTERS

REVIEWS

Chapter 1: I'm Delusional?

Chapter 3: I Know

Chapter 2: I Befriended the "System"

Chapter 4: A Special Signal

<

1

>

AZ ↑

Figure I: Novel description page

CHAPTERS

REVIEWS

AZ ↑

Rate this novel

Tell us what you think.

☆☆☆☆☆

SUBMIT REVIEW

RA rachael

8/10/2022, 11:44:39 AM

Not too big of a problem considering the previous chapter lengths made it a bit tedious to read. Plus the cut-offs make sense, as opposed to how "Damn reincarnation" horribly splits chapters (killing the pacing)

[More](#)

★★★★☆ 3 stars

PH phanindrapanta

8/9/2022, 12:55:34 PM

It's competently written but there's very little in that will elicit tension or investment.

This story is the equivalent of eating unseasoned mashed potatoes: there's nothing terribly wrong with it, but there's not much that's actively good either....

[More](#)

★★★★★ 5 stars

NI nishan

8/9/2022, 12:54:58 PM

Welcome to the College of Winterhold... i mean the Lyceum, where our budding MC learns about the Destructive arts, with a side of Alteration.

Right off the hop the author starts flushing out his magic world showing the difference between the haves and the have nots in tha...

[More](#)

★★★★☆ 3.5 stars

<

1

>

Figure J: Review section in novel description page

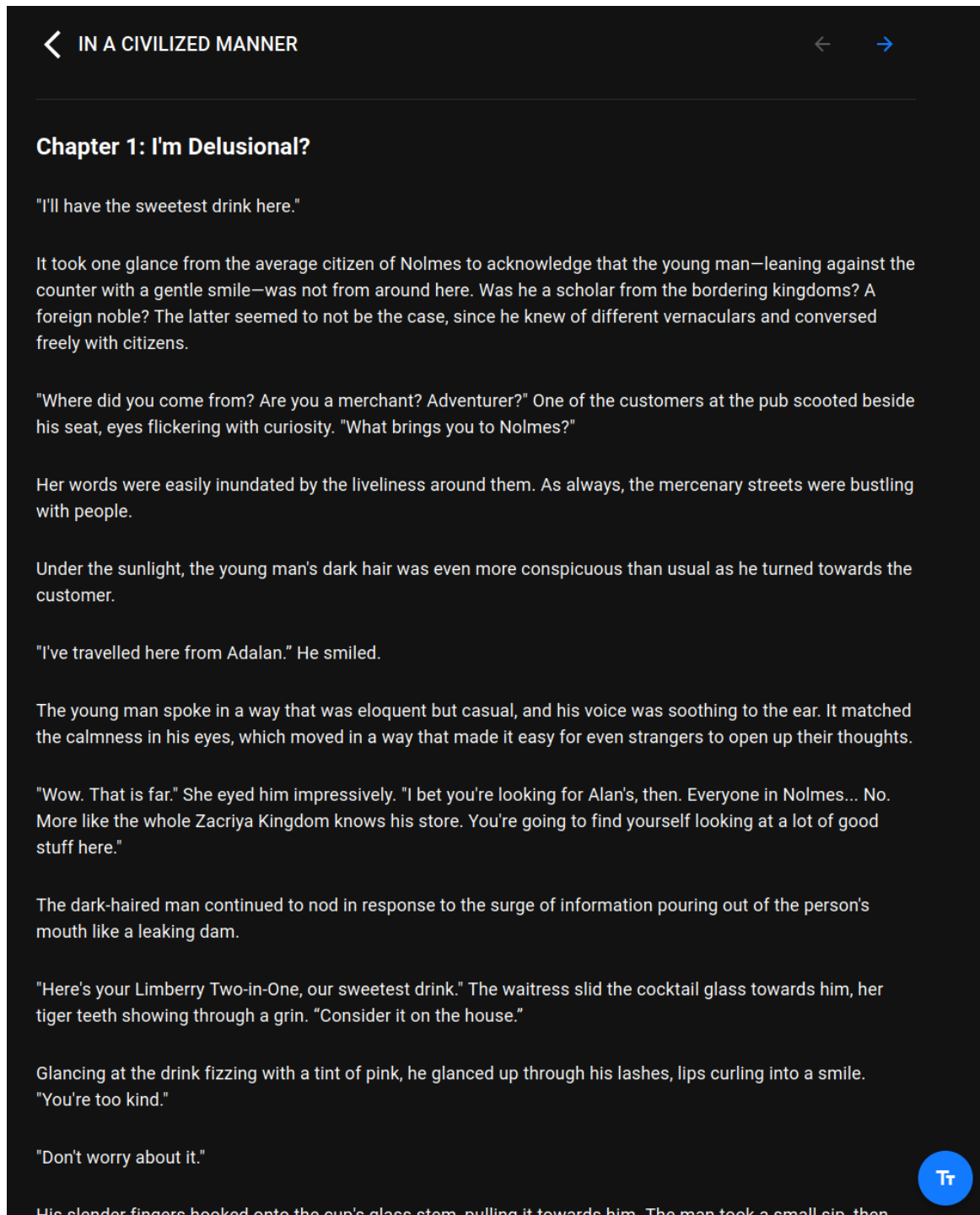


Figure K: Page displaying chapter content along with speed dial for changing font size

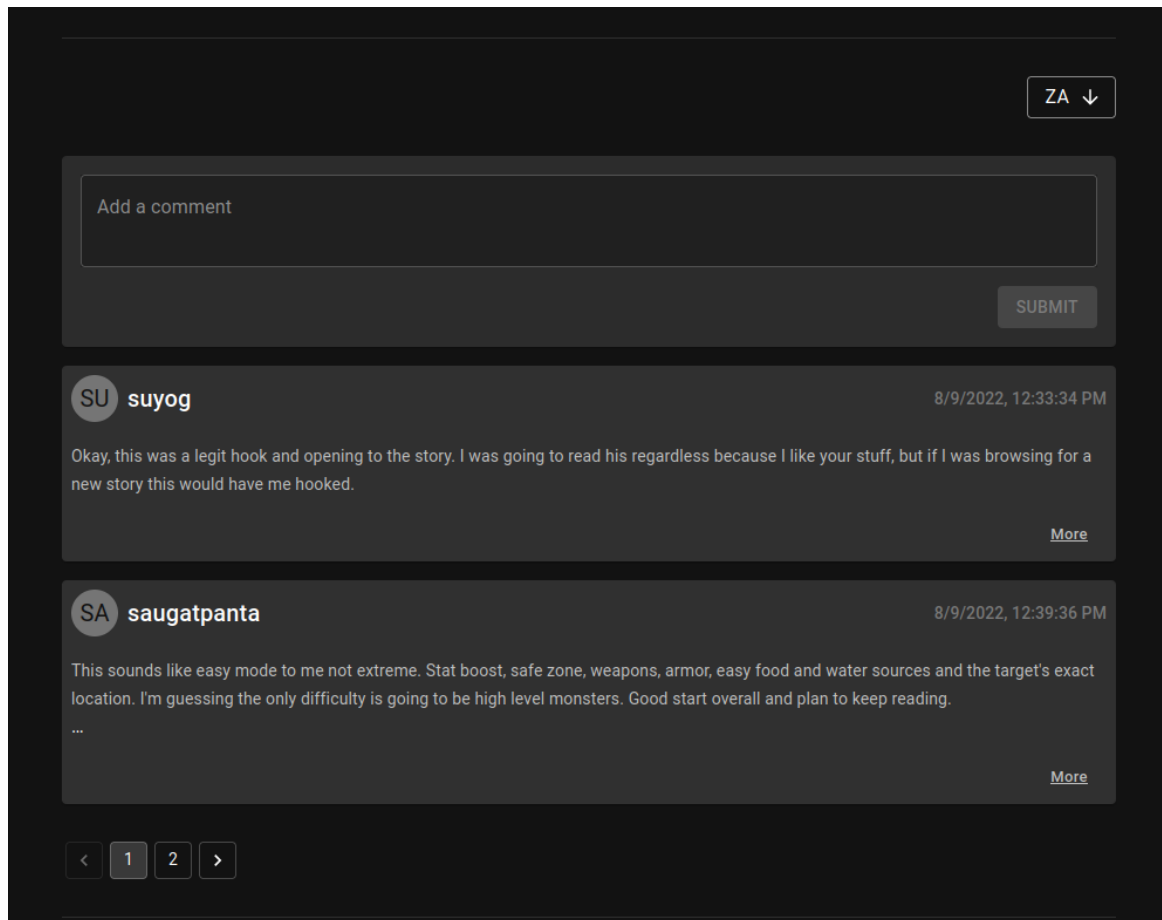


Figure L: Comment section

Status

☐ Any
☐ Ongoing
☒ Completed
☐ Hiatus

Sort By

☐ Name
☐ Popular
☐ New
☒ Rating

Select Genres

Mature
Sci Fi
Add Genre

CLEAR

APPLY FILTERS

Completed

Demon Hunter

★★★★★ 5.0 stars

Following the nuclear warfare, massacre extended towards the entire world. The order of mankind completely collapsed, and survival of the fittest became the ultimate principle....

Sci Fi
Action
Tragedy
Dystopian

Completed

Earthside Consortium

★★★★☆ 4.0 stars

Gaston Hardy is a Freelancer currently under the supervision of the United Earth Defense Force. He acts as a Courier, Messenger, and Adjutant to those who acquires his services...

Sci Fi
Action
Thriller
True Crime

Completed

Deadman

★★★☆☆ 2.3 stars

Even after the bombs fell, even after countries unleashed weapons beyond imagining on one another, even after lawlessness became the norm, people still expect their mail...

Sci Fi
Dystopian

Figure M: Series page with filtered list of novels

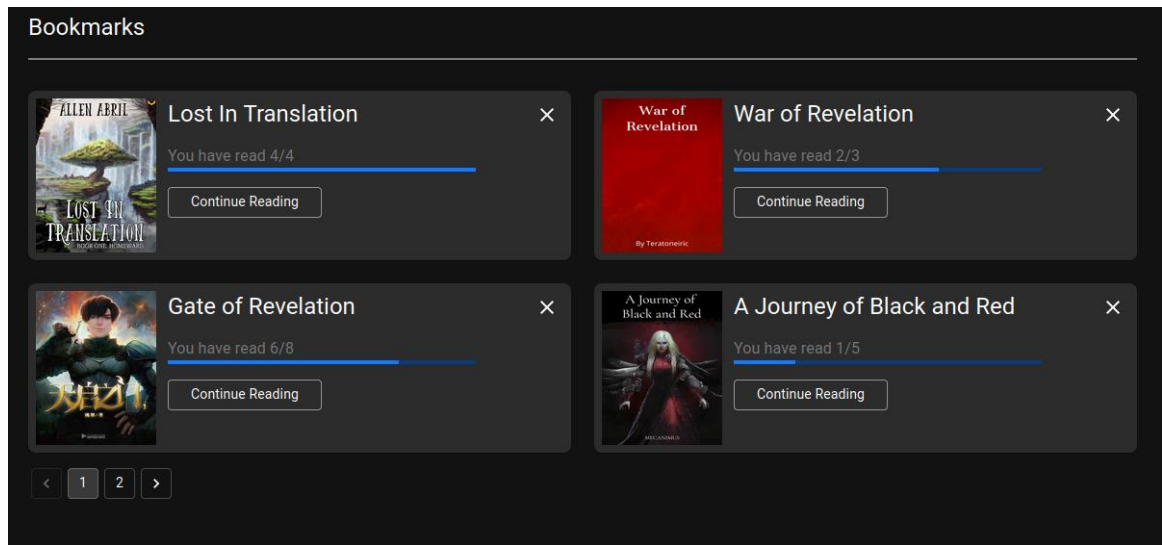


Figure N: Bookmarks page with bookmarked chapters














Recently Updated		
Title	Release	Time
 Falling with Folded Wings	Chapter 5 : M5	2 days ago
 Falling with Folded Wings	Chapter 4 : M4	2 days ago
 Falling with Folded Wings	Chapter 3 : M3	2 days ago
 Falling with Folded Wings	Chapter 2 : M2	2 days ago
 Falling with Folded Wings	Chapter 1 : M1	2 days ago
 139: In Evening	Chapter 5 : The Last Good Day	2 days ago
 139: In Evening	Chapter 4 : The Last Good Day	2 days ago
 139: In Evening	Chapter 3 : Somnidin	2 days ago
 139: In Evening	Chapter 2 : Normal	2 days ago
 139: In Evening	Chapter 1 : Yesterday, Tomorrow, Today	2 days ago
 Deadman	Chapter 6 : One by One	2 days ago
 Deadman	Chapter 5 : The Long Machete of the Law	2 days ago
 Deadman	Chapter 4 : A Boon for Boon	2 days ago

Figure O: List of recently updated chapters in home page