

# KUEBIKO

## PROJECT REPORT

### Submitted By-

Padm Keshav	20BCE2708
Anushka Srivastava	20BCE02033
Rudhradeep Biswas	20BCE0683

CSE 3001 Software Engineering (EPJ)

### Project Guide

**Instructor: Prof. Swarnalatha P**

Associate Professor Grade 1

School of Computer Science and Engineering

## BACHELOR OF TECHNOLOGY

in

## COMPUTER SCIENCE AND ENGINEERING



**VIT<sup>®</sup>**  
**Vellore Institute of Technology**  
(Deemed to be University under section 3 of UGC Act, 1956)

## SCHOOL OF COMPUTER SCIENCE AND ENGINEERING

Winter Semester, 2021 - 2022

## **TABLE OF CONTENTS**

<b>ABSTRACT</b>	<b>4</b>
<b>INTRODUCTION:</b>	<b>5</b>
Motivation:	5
Aim of the Proposed Work:	5
Objective:	5
Report Organization:	5
<b>Literature Survey:</b>	<b>6</b>
Survey of Existing Models/Work	6
Summary/Gaps identified in the Survey	7
<b>Proposed System Requirements Analysis and Design:</b>	<b>9</b>
Introduction	9
Requirement Analysis	9
Stakeholder Identification	9
Functional Requirements	9
Non-Functional Requirements	12
System Requirements	13
H/W Requirements	13
S/W Requirements	13
Software Requirement Specification document	14
Work breakdown structure	14
Gantt Chart	15
<b>Design of the Proposed System:</b>	<b>15</b>
Introduction	15
Benefits of Proposed System	15
High Level Design	16
Architecture Design	16
Architecture Diagram	17
Detailed Design	17
ER diagram	17
Database Schema	18
UML diagram	19
<b>Implementation and Testing:</b>	<b>25</b>
Implementation details	25
Testing	34
<b>Conclusion, Limitations and Scope for future Work:</b>	<b>36</b>
<b>References:</b>	<b>37</b>
List of Websites	37
List of Books	37

## **ABSTRACT**

This website will act as a platform for the writers to showcase their creativity and the language will not be limited to just English. Currently people who cannot enjoy their local novels due to inaccessibility or unavailability. The users will be able to get content from their favorite authors.

The TTS (Text to Speech) feature can also be used by the readers who have visual imparity. This also helps in providing the users with the hands-free feature so they canenjoy while doing their other work simultaneously.

# **1. INTRODUCTION:**

## **1.1. Motivation:**

Although there are many novel reading and writing sites, there is no site that caters to the non-English readers and visually impaired readers. We plan on including these readers as well as normal readers into our user base. There is also an increased need for a platform for Indian novel writers to express their talents in either English or their regional languages.

## **1.2. Aim of the Proposed Work:**

The aim of our project is to create an inclusive Novel reading and writing software in which readers can read through a wide variety of novels of different genres and is available in different languages and also, writers can exhibit their writing skills and upload their own work for others to read and get recognized through our website. The website can be applicable to people of all age groups as they can access and read books of their choice. This website also acts as a platform for new and rising authors to write and share their books for the public to view and gain popularity. This will act as a source of motivation for them to keep writing and improving.

## **1.3. Objective:**

The main objective of the project is to provide a platform for its users to read their favorite novels and also for the authors to showcase their creativity.

We plan on implementing text to speech features so that visually impaired users can also enjoy the content of the books.

The website enables the user to upload books in multiple languages thus people from different regions can upload and read books.

## **1.4. Report Organization:**

The report is organized into 4 parts. The first part introduces the project to the reader. Its main purpose is to provide an overview of the project. The second part shows the necessary research and requirement analysis done for the project. It is followed by the third part where the design decisions and design

specifications are shown to the reader. The final part shows the implementation details, the final actual design of the software and tests conducted on the software.

## **2. Literature Survey:**

### **2.1. Survey of Existing Models/Work**

#### **Existing Models**

Webnovel - Webnovel is an English Novel reading and writing website. It primarily focuses on translated novels from its parent website Qidian - a Chinese regional Book reading and writing website.

Wattpad - Wattpad is an English Novel reading and writing website.

Pratilipi - It is an Indian Novel reading and writing website. It provides various novels in various Indian native languages for native users to read.

Audible - It is a very popular audiobooks website

#### **Research Papers**

##### **A review and analysis of technologies for developing web applications**

Asha Mandava and Solomon Antony Murray state University Murray, Kentucky

##### **The Impact of E-Books on the Printed Books: E-book Popularity, Growth and future**

Conference: Fifth International Conference on e-Learning entitled "Cognitively informed technology" TheAt: Manama, Bahrain

##### **Comparison of AngularJS and React.js frameworks based on a web application**

March 2018, Journal of Computer Sciences Institute

## **2.2. Summary/Gaps identified in the Survey**

### **Existing Models**

Webnovel - Focuses mainly on Chinese books translated to English. Not much support for multiple languages.

Wattpad - English only site that does not support any other language. No support for visually impaired users or non-English users.

Pratilipi - Heavy focus on native Indian languages. No support for interchangeability of languages. No support for visually impaired users.

Audible - Great application for visually impaired users through the use of audiobooks. Not much support for normal book reading, No support for independent on-site book writing.

### **Research Papers**

#### **A review and analysis of technologies for developing web applications**

Asha Mandava and Solomon Antony Murray state University Murray, Kentucky

This paper focuses on technologies useful for design and development of web-based applications and technologies used at the client side and server side of web applications. It gives an overview on how different frameworks can be used hand in hand for better results. It gave an overview on object-oriented approach in web development and its advantages. But there was no mention of newer and rising technologies that can be helpful for web development in the long run.

#### **The Impact of E-Books on the Printed Books: E-book Popularity, Growth and future**

Conference: Fifth International Conference on e-Learning entitled "Cognitively informed technology" TheAt: Manama, Bahrain

This study aims to identify the preferences of the net generation readers in print and electronic media activities and to investigate the impact of e-books on the printed books, and try and see if printed books are being gradually replaced by electronic one. According to the research, the presence of books will be reduced drastically within a few years and most work will be done with eBooks, but that is very theoretical. Practically it is very hard to get eBooks of all the hardcopies available. And also, it might

be harder to get a softcopy than to get out hands on a hardcopy of some locally available books.

### **Comparison of AngularJS and React.js frameworks based on a web application**

March 2018, Journal of Computer Sciences Institute

This survey's main purpose was to find which front end development framework would be best for implementation on our website. Since both the frameworks are pretty famous and also widely used, both can have different advantages/disadvantages on implementation of the application. This survey helped us to get into depth of the above listed frameworks and let us finalize one.

### **3. Proposed System Requirements Analysis and Design:**

#### **3.1. Introduction**

We plan on developing a website that acts as a platform to users from various regions to access content in various languages and also regional authors as well as international authors provide their content to the users, and we plan to keep the website open source and the content available will be free of cost hence the users won't have to pay for any book they read or they decide to submit

#### **3.2. Requirement Analysis**

##### **3.2.1. Stakeholder Identification**

There are 2 types of end users: Readers and Writers.

Writers:

Share their work and creation to the public. Showcase their skills inexpressing emotions in words.

Readers:

These are the people who are going to read the books for their satisfaction and hobbies. People with visual imparity can also make use of the TTS to enjoy creations from their favorite authors without being dependent on anyone.

##### **3.2.2. Functional Requirements**

The different Functional Requirements for our Project are given below.

#### **1. Login**

##### **1.1 Description and Priority**

This allows the User to save their progress, their search history and allows them to rate and comment on any book. It is of high priority as it's important for the user to be logged in to the webapp.

#### **Stimulus/Response Sequences**

**Stimulus:** User identify themselves to the display manager or login with a login name and password.

**Response:** That login name is looked up.



**Stimulus:** The user enters an account password which is encrypted in exactly the same way.

**Response:** The login program checks to see if they match.

**Stimulus:** The user has successfully logged in.

**Response:** They will be able to access all the features associated with the individual account they are using.

### **Functional Requirements**

#### ***Check credentials***

**Input:** Username and Password

**Output:** Logged in if credentials are correct; Error screen if incorrect

**Processing:** Search Username in database. If none found redirect to sign up. If found, check password. If matched, successfully log in User.

## **2 Sign Up**

### **2.1 Description and Priority**

This allows the User to create their own account on the site. It is of high priority as a user can't access the webapp without signing it to the db.

#### **Stimulus/Response Sequences**

**Stimulus:** User wishes to create their own account on the website to access the features offered.

**Response:** The website gives a prompt to the user to enter details.

**Stimulus:** The user creates a user name and an account password which is encrypted in exactly the same way.

**Response:** The program stores the data in the database for future login.

**Stimulus:** The user has successfully created the account and is now legible to use all the feature provided by the website.

**Response:** They will be able to access all the features associated with the individual account they are using.

### **Functional Requirements**

#### ***Input credentials to database***

**Input:** User details and

Password

**Output:** Add user to database. Display Sign Up successful.

**Processing:** Add User details according to schema to User Database.

### **3. Log Out**

#### **3.1 Description and Priority**

This allows the User to log out of their own account on the site on the device that they are currently using. It is of low priority as a user mostly prefers saving his/her login credentials thus making logging out of the webapp not a function of much importance.

#### **Stimulus/Response Sequences**

**Stimulus:** User wishes to logout and end access to the WebApp. User clicks on the logout button.

**Response:** The website gives a prompt to the user to logout and exit. Users' data and progress is saved and the user is logged out.

#### **Functional Requirements**

##### ***Input credentials to database***

**Input:** “Logout” option is clicked.

**Output:** Display Logout successful.

**Processing:** Save all current data of user to database. Remove all processes going on for the current user on the device.

## **Functional requirements primarily for Readers are given below.**

### **4. Search for Books**

#### **4.1 Description and Priority**

This allows a user to search a particular book or author by keywords and find them. If multiple results are found, the user can search according to specific filters. It is of medium priority as searching for books can be taken under the section of all books.

#### **Stimulus/Response Sequences**

**Stimulus:** User wishes to look up a book or an author. The user enters the keyword in the search bar.

**Response:** The phrase entered by the user goes into the database and the website returns a list of what it thinks is the most relevant pages to the user's search.

**Stimulus:** User selects the choice which they were looking for.

#### **Functional Requirements**

##### ***Search for keyword***

**Input:** Keywords

**Output:** List of Books, Authors, Tags or Genres that match the keywords

**Processing:** Search keywords in the Book database. If keywords match Book name, Author name, Tag name or genre name, add to list. Finally display it.

##### ***Search using Filter***

**Precondition:** Search for keywords done.

**Input:** Filter option

**Output:** List of Books, Authors, Tags or Genres that match the keywords

**Processing:** If List contains option specified in Filter keep it else remove it from list.

##### ***Sort Results***

**Precondition:** Search for keywords done.

**Input:** Sort option

**Output:** List of Books, Authors, Tags or Genres that match the keywords

**Processing:** Sort the list according to the option specified.

## **5. View Book Details**

### **5.1 Description and Priority**

This allows the user to see what the various details of the book are. The user clicks either on the title or the book cover and gets redirected to the Book details page. It is of low priority as a user would be more inclined to the contents of the book having known the book details most probably beforehand.

#### **Stimulus/Response Sequences**

**Stimulus:** User wishes to look up details of a book. The user clicks on the book details button.

**Response:** The website redirects the user to the book meta data containing the title, subtitle, author's name, series name, book description, and all.

#### **Functional Requirements**

##### ***Open Book Details***

**Input:** BookID of the Book that was clicked

**Output:** Page that contains details of the book.

**Processing:** Search database for bookID, find it and display it.

## **6. View Contents of a Chapter**

### **6.1 Description and Priority**

This allows the user to get a brief idea on the number of chapters present in the novel. They can get a summary of what happens in that chapter before they dive into the actual content. It is of high priority as it gives user the actual content present in the book in a more concise manner so that he/she can navigate accordingly inside the book.

#### **Stimulus/Response Sequences**

**Stimulus:** User wishes to look up meta data of a particular chapter. The user clicks to show more of contents of a chapter.

**Response:** The website returns brief meta data of each chapter, such as summaries of each chapter, number of pages, etc.

#### **Functional Requirements**

##### ***Chapter Content***

**Input:** Chapter number of the Book that was clicked

**Output:** Page that contains chapter details.

**Processing:** Search database for chapter number, find it and display it.

## **7. Change Language**

## 7.1 Description and Priority

This allows the user to change language to their native language. User clicks on the language from a list of languages. It is of high priority as it makes native language speakers to enjoy the book contents in his/her preferred language making it a more holistic experience for the user.

### Stimulus/Response Sequences

**Stimulus:** User wishes to change the language of the novel content. The user selects on the drop- down box to select their preferred language.

**Response:** The website gives all the available choices to the user in the drop-down.

**Stimulus:** User selects their preferred language.

**Response:** The text is converted to user's desired language.

### Functional Requirements

#### *Change Language*

**Precondition:** Book and Chapter number has been selected.

**Input:** LanguageID of the Language that was clicked

**Output:** Content of the chapter in the language specified

**Processing:** Search database for bookID for the specified languageID, find it and display it. If not found, convert contents to the specified language using machine translation.

## 8. Toggle Text to Speech

### 8.1 Description and Priority

This allows the user to activate the text to speed feature that can read out the content of the book. This is helpful for people with visual impairment. It is of high priority as this feature is one of the trademark features of this webapp where even the person facing visual impairment can read the book which makes the webapp for everyone no matter what.

### Stimulus/Response Sequences

**Stimulus:** The user is visually impaired and wishes to read a certain book. They click the volume button to enable text to speech feature.

**Response:** Text to speech feature is enabled.

### Functional Requirements

#### *TTS Details*

**Input:** TTS button is clicked.

**Output:** Voice that reads out the content of the book line by line.

**Processing:** Activate the TTS feature from the server/ api used

## **9. Rate a Book**

### **9.1 Description and Priority:**

This allows the user to give ratings to the book that they read so that others can get to know the statistics of the popularity and thoughts of other users. It is of low priority as rating a book is user to user preference and also the user most likely know how a book is before diving in to read it.

### **Stimulus/ResponseSequences**

**Stimulus:** User rates a book after reading it.

**Response:** The rating is made public.

**Stimulus:** Other users get familiar with the popularity of the book.

### **Functional Requirements**

#### ***Rating Details***

**Input:** Rating section is selected

**Output:** User allowed to input the number of stars that they think the book deserves

**Processing:** Get request from server, after user gives the rating add it to the database and display the overall rating

## **10. Write a Comment**

### **10.1 Description and Priority:**

This allows the user to provide short comments to a book. It can be a helpful guide for other users who wish to read the book. It is of medium priority as user comments can act as a suggestion for us the creators to improve the webapp and update them according to the needs of our users.

### **Stimulus/Response Sequences**

**Stimulus:** Users share their views about a novel read by them allowing other users to know a gist about the novel before they read it.

**Response:** The review is stored and made public to other users.

**Stimulus:** Users share their reviews about the novel read directed towards the creators and authors.

**Response:** The review is stored and made accessible to the author or creator.

**Stimulus:** The creator or author reads the reviews helping them better the quality of their work and produce novels which cater to user needs.

**Stimulus:** Users share the issues or glitches that occurred in the functioning of the program while the user was reading a specific book.

**Response:** The review is stored and help developers to trouble shoot the problem with the program.

### **Functional Requirements**

#### ***Write Comment***

**Input:** Heading and content of comment.

**Output:** Display comment added and add comment to database.

**Processing:** Checks for BookID of the selected book and adds the comment to its Feedback module.

**Functional Requirements primarily for writers are as follows:**

### **11. Create a New Book.**

#### **Description and Priority:**

Any user that wants to write/ submit a book can select this option. The user is led to a page where they can give their input and submit, after which it is published on the website. It is of high priority as it is one of the webapp's main section where any user can create his own books making it reach out to the world and our users.

#### **Stimulus/Response Sequences**

**Stimulus:** Users wishes to try their hand at writing their own novels and share it with other users.

**Response:** The application enables this feature and allows the user to write and post the content under their name.

### **Functional Requirements**

#### ***Book Creation Details***

**Input:** Details of the new book.

**Output:** Display new book created.

**Processing:** Input new book data into the database.

### **12. Add a New Chapter**

### **12.1 Description and Priority:**

This allows the user to add a new chapter to an already existing book. It is of high priority as after creating the book it is the most basic function to be performed that is adding chapters to it otherwise it's of no use.

#### **Stimulus/Response Sequences**

**Stimulus:** Novel written by an author or a user is still ongoing and they wish to update their novel.

**Response:** The application enables the author to write a new chapter and upload it. Other users who have bookmarked this novel receive a notification of this update.

#### **Functional Requirements**

##### ***Write a New Chapter***

**Precondition:** User has created a book.

**Input:** Heading and content of chapter.

**Output:** Display Chapter added and add chapter to database.

**Processing:** Checks for BookID of the selected book and adds the chapter to its Content module.

### **14. Delete Chapter**

#### **14.1 Description and Priority:**

This allows the user to delete any chapter that they have published in the past. They are prompted by the system to confirm if they want to delete their content and after confirming it is deleted. It is also of high priority as it gives the user the freedom to experiment with the contents of the book knowing the content can be deleted afterwards.

#### **Stimulus/Response Sequences**

**Stimulus:** An author wishes to delete a certain chapter from their novel.

**Response:** The application enables the author to delete the desired chapter and updates the database.

#### **Functional Requirements**

##### ***Chapter Delete***

**Input:** Delete chapter button clicked.

**Output:** User asked if they want to confirm the request, on confirm delete the chapter.

**Processing:** Delete request sent to server, prompt presented, deleted from database.

### **15. Delete Book**



### 15.1 Description and Priority:

This allows the user to delete a book they have submitted in the past. It is of medium priority as the database is strong enough from the creator side to hold on to un-deleted books but it can crash if a system overload happens, hence users would be encouraged to delete the book if they don't intend to keep it in the webapp.

#### **Stimulus/Response Sequences**

**Stimulus:** An author wishes to delete a certain novel written by them.

**Response:** The application enables the author to delete the desired novel and updates the database.

#### **Functional Requirements**

##### ***Select Book***

**Input:** BookID to be deleted

**Output:** Select Book

**Processing:** Search for bookID in database

##### ***Delete Book***

**Input:** Confirmation for deletion

**Output:** Display Book deleted.

**Processing:** Delete all chapters for the bookID. Delete all feedback for the bookID. Delete the details of the book with the bookID.

### 3.2.3. Non-Functional Requirements

#### **Performance Requirements**

- The time to load the landing page should be less than 4 seconds.
- The response time for browsing and searching should be less than 5 seconds.
- The time taken to update the database or to retrieve information from the database should be less than 4 seconds.
- The webapp shall handle expected and non-expected errors in ways that prevent loss in information and long downtime period. Thus, it should have inbuilt error testing to identify invalid username/password.
- The system should be able to handle large amount of data. Thus, it should accommodate high number of books and users without any fault.

#### **Safety Requirements**

The database may get crashed at any certain time due to virus or operating system failure. Therefore, it is required to take the database backup so that the database is not lost. Proper UPS/inverter facility should be there in case of power supply failure.

#### **Security Requirements**

- System will use secured database.
- Normal users can just read information but they cannot edit or modify anything except their personal and some other information.
- System will have different types of users and every user has access constraints.
- Proper user authentication should be provided.
- No one should be able to hack users' password.
- There should be separate accounts for admin and members such that no member can access the database and only admin has the rights to update the database.
- The users should keep their system access credentials to themselves in order to prevent unauthorized access and misuse of data.
- If at any point a user decides to discontinue their services, the system should allow them to do so after proper authorization and proof.
- In case the user forgets/ misplaces the key to login into their account they should be provided with the facility to recover their account or in the worst case if the account is not recoverable, terminate the account.

### **Software Quality Attributes**

- The prioritization of the software quality attributes is assumed as under:
- Accurate and hence reliable
- Secured
- High performance
- Compatibility

### **Business Rules**

A business rule is anything that captures and implements business policies and practices. A rule can enforce business policy, make a decision, or infer new data from existing data. This includes the rules and regulations that the System users should abide by. This includes the cost of the project and the discount offers provided. The users should avoid illegal rules and protocols. Neither admin nor member should cross the rules and regulations.

### **3.2.4. System Requirements**

#### **3.2.4.1. H/W Requirements**

The hardware interface for the user would be any PC having a configuration of P-IV and above 2GB HDD for loading any OS so that it can interact with the system without any problem. The main interface would be monitor, keyboard and mouse.

#### **3.2.4.2. S/W Requirements**

Kuebiko would take help of a database for storing and managing the novels being uploaded on the website by the authors. So, an access to the database management system is required. When such an event occurs, the system establishes connection to the database management system; once the connection is created; the client program can communicate with the database management system.

### **Softwares Used:**

**Editor:** Vscod

**Programming Languages Used:**

HTML

CSS

JavaScript

### Front End Frameworks Used:

React.js

Bootstrap

### Backend Frameworks Used:

Node.js

MongoDB

Mongoose

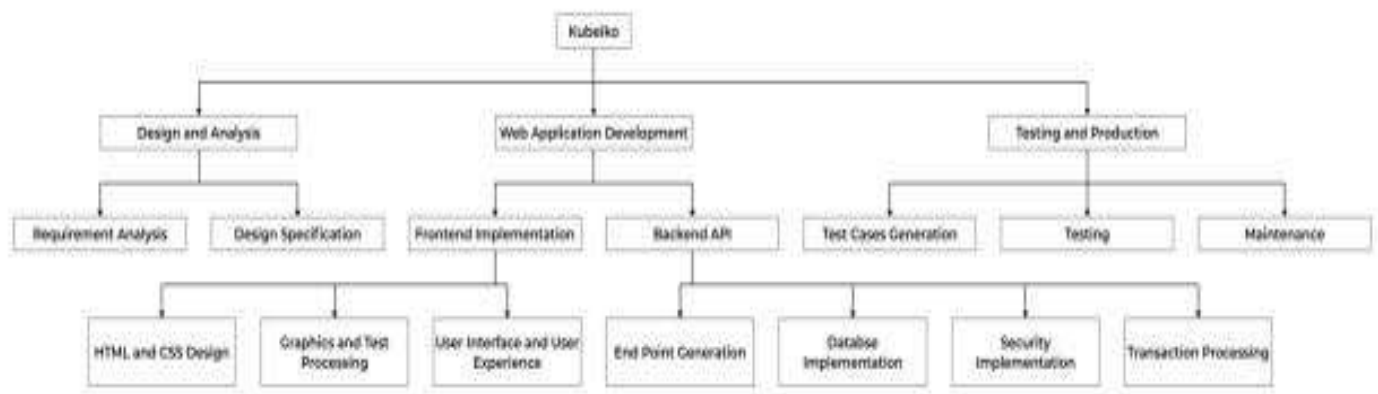
Passport.js

### 3.2.5. Software Requirement Specification document

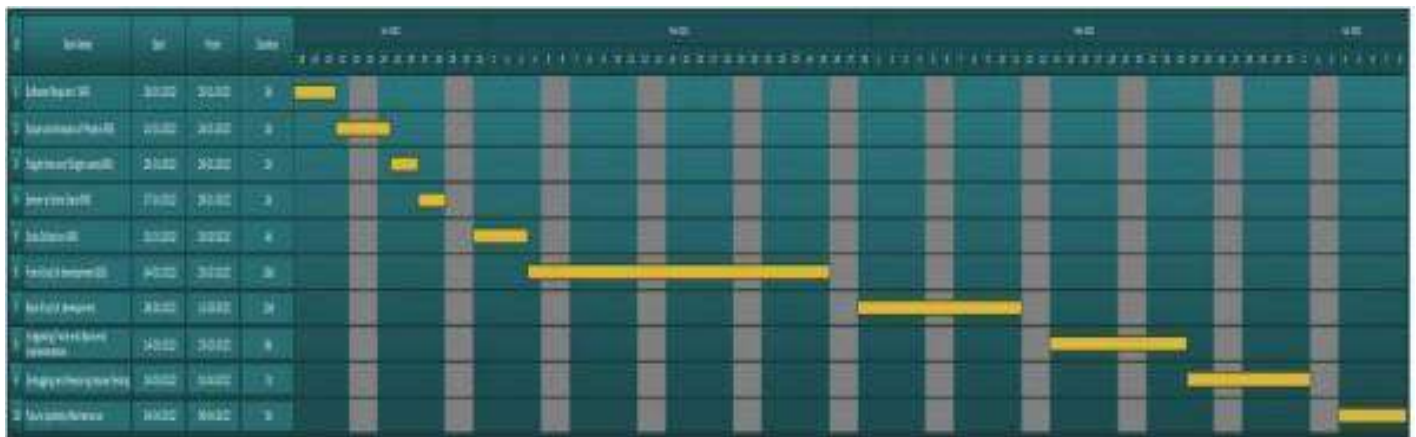
The Software Requirement Specification (SRS) document is available here.

[https://drive.google.com/file/d/16PAmg8ovQS2oAA5r736UtHx\\_xkDieCUx/view?usp=sharing](https://drive.google.com/file/d/16PAmg8ovQS2oAA5r736UtHx_xkDieCUx/view?usp=sharing)

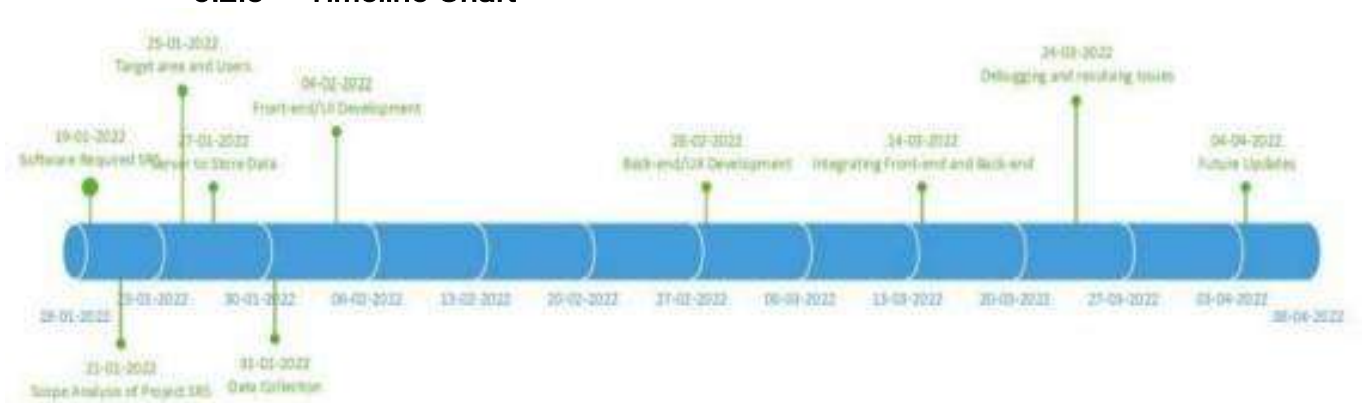
### 3.2.6. Work breakdown structure



### 3.2.7. Gantt Chart



### 3.2.8 Timeline Chart



## 4. Design of the Proposed System:

### 4.1. Introduction

We plan on developing a website that acts as a platform to users from various regions to access content in various languages and also regional authors as well as international authors provide their content to the users, And we plan to keep the website open source and the content available will be free of cost hence the users won't have to pay for any book they read or they decide to submit.

### 4.2. Benefits of Proposed System

- There are a few websites that also have the same purpose as our website but most of them have premium content for which the users have to pay in order to access.
- We have implemented a Text to speech feature in our feature to our website which is a very new concept and can be helpful for visually impaired people or people can use this feature to enjoy their content while they do their work.
- Since there is no restriction in the language of content being posted, users from different regions can enjoy reading their books as well as regional

authors can also share their content with everyone.

## 4.3. High Level Design

### 4.3.1. Architecture Design

- MVC Architectural Style

MVC (Model - View - Controller) style separates presentation and interaction from the system data. The system is structured into 3 logical components that interact with each other.

Model - This component manages the system data and associated operations on that data.

View - This component defines and manages how the data is presented to the user.

Controller - This component manages user interaction and passes these interactions to the View and Model components.

We will use this MVC Style for our project Kuebiko because there are multiple ways to view and interact with data. We will use this to avoid the problem where due to intermixing of the code between the process logic and interface, the modularity of the application is reduced and it becomes difficult to maintain. The main advantage of this style is that it allows the data to change independent of its representation and vice versa.

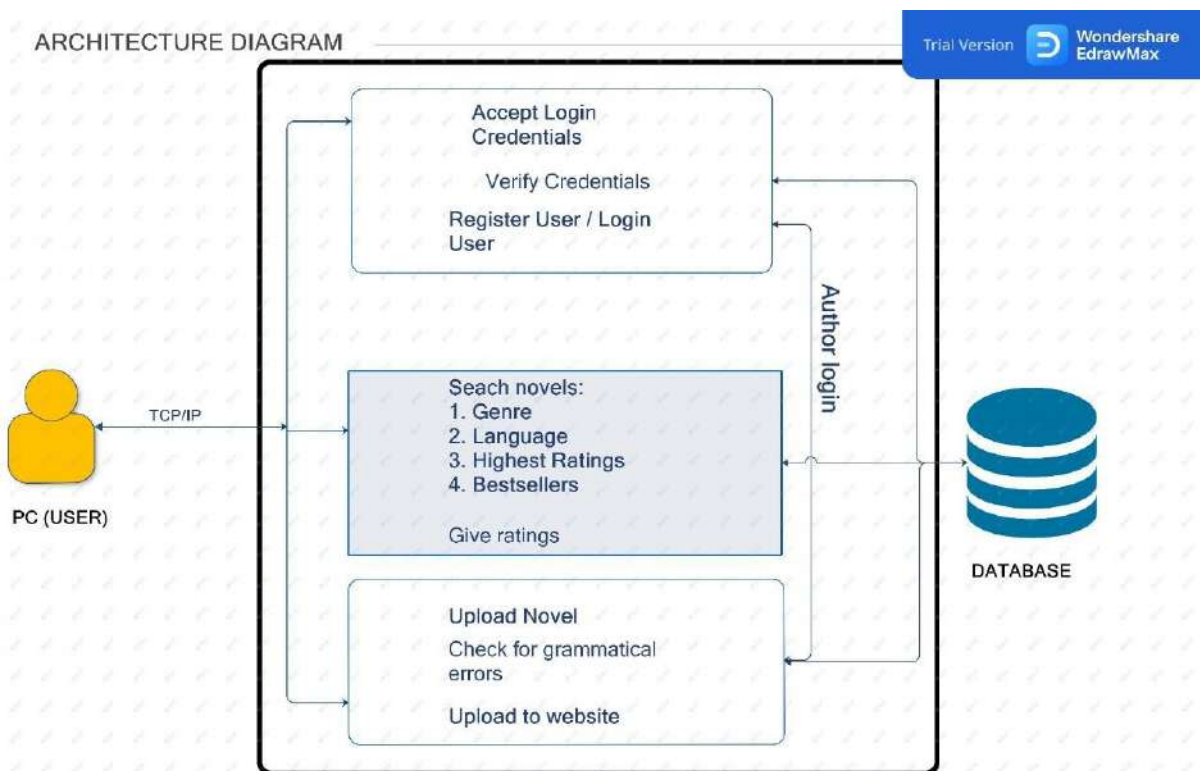
- Three-tier Client Server Architecture

In a client server architecture, the functionality of the system is organized into services, with each service delivered from a separate server. Clients are users of these services and access servers to make use of them.

We will use this 3-tier Client Server Architecture because, when data in a shared database has to be accessed from a range of locations, this architecture performs the best. Because the servers can be replicated, they can be increased when the load on a system is high and decreased when the load reduces.

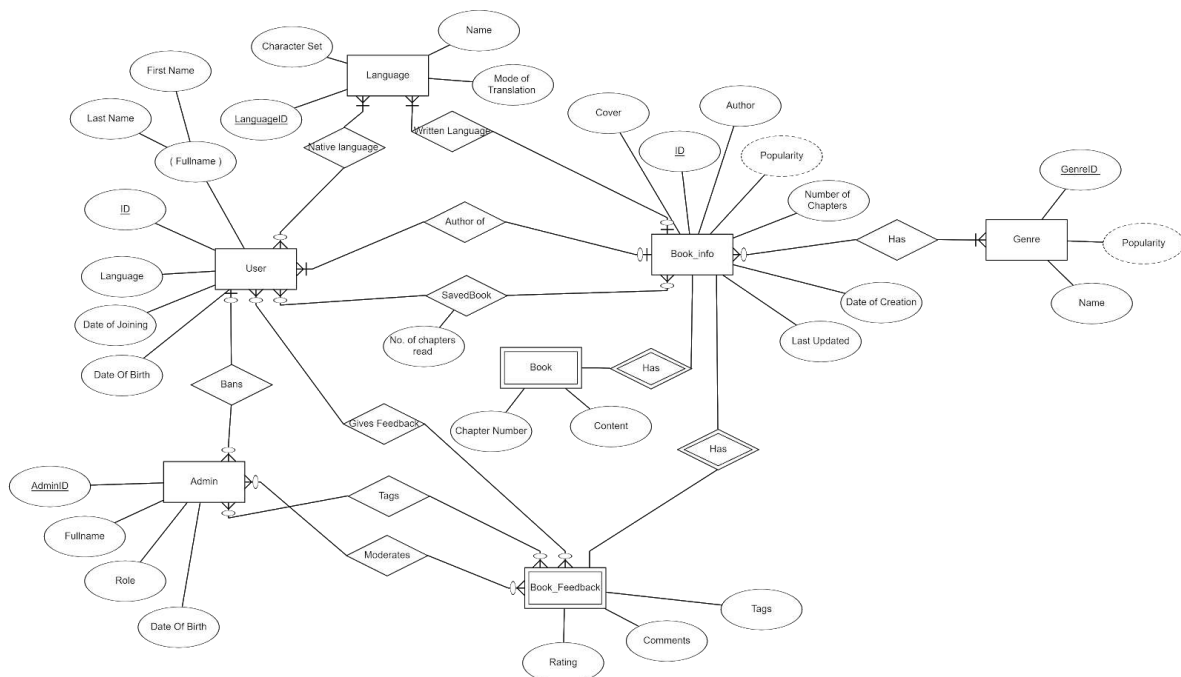


### 4.3.2. Architecture Diagram



## 4.4. Detailed Design

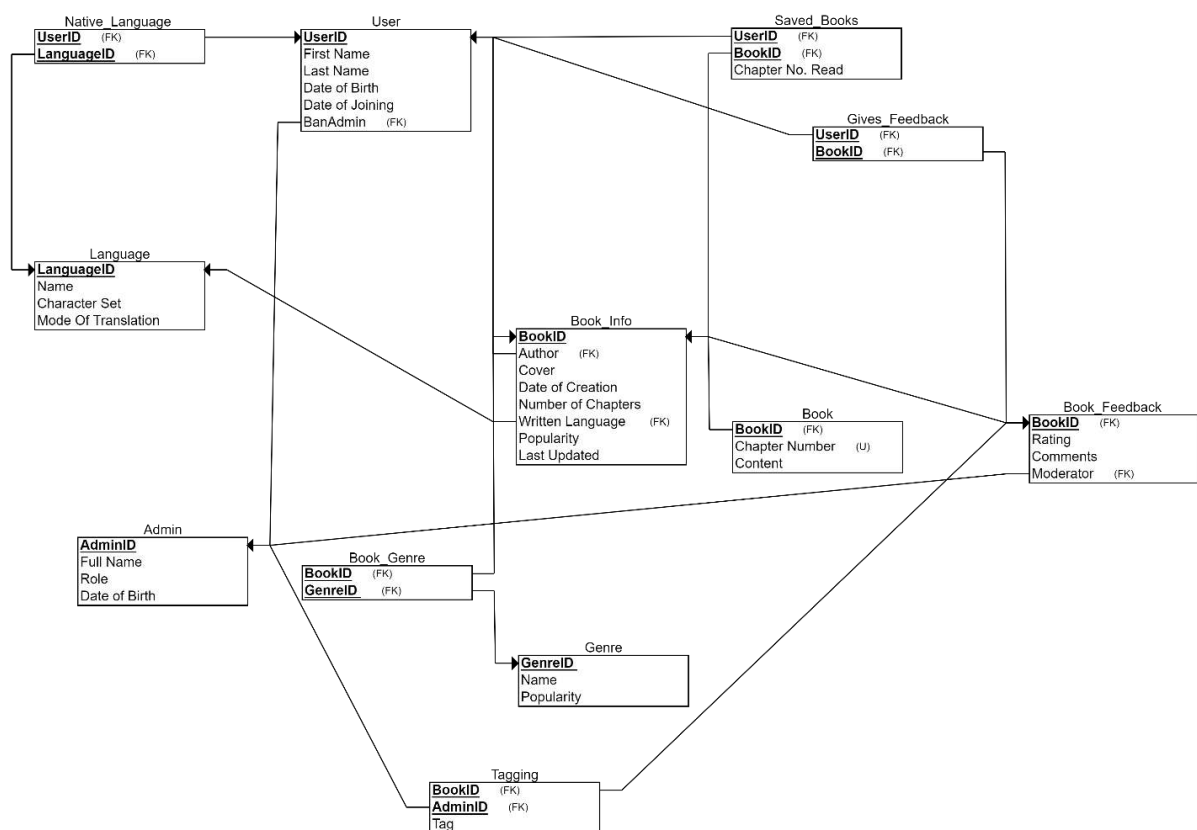
### 4.4.1. ER diagram



The ER diagram helps us in understanding how the data is stored in the database. The user entity contains the user id , email and other information that can be used during authentication. The books can be accessed by using the specific book id which is the unique key that distinguishes different books. And the books can be divided into different genres and the genreID helps in filtering the books into different genres. The languageID is used when the author decides to submit books of different languages. This helps in differentiating regional books from english books. When the user creates a book a new book with an unique bookID is created.

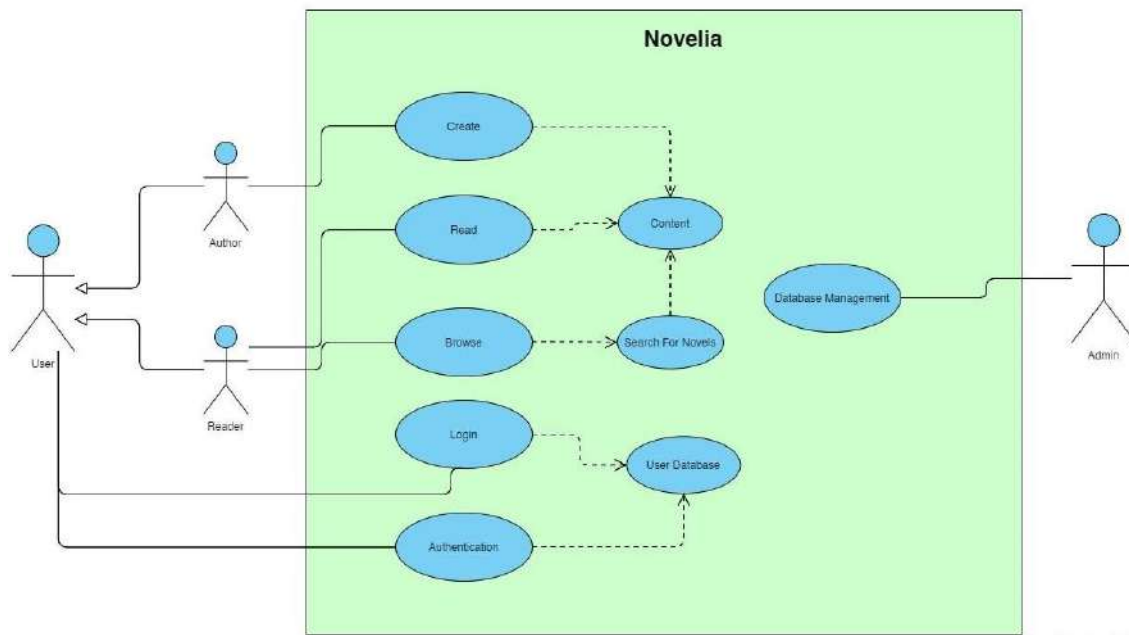
The Database schema is a conceptual blueprint of our database which outlines the way our database is set up and will possibly be used. Based on our ER Diagram we have finalized a Database relational schema as follows:

#### 4.4.2. Database Schema

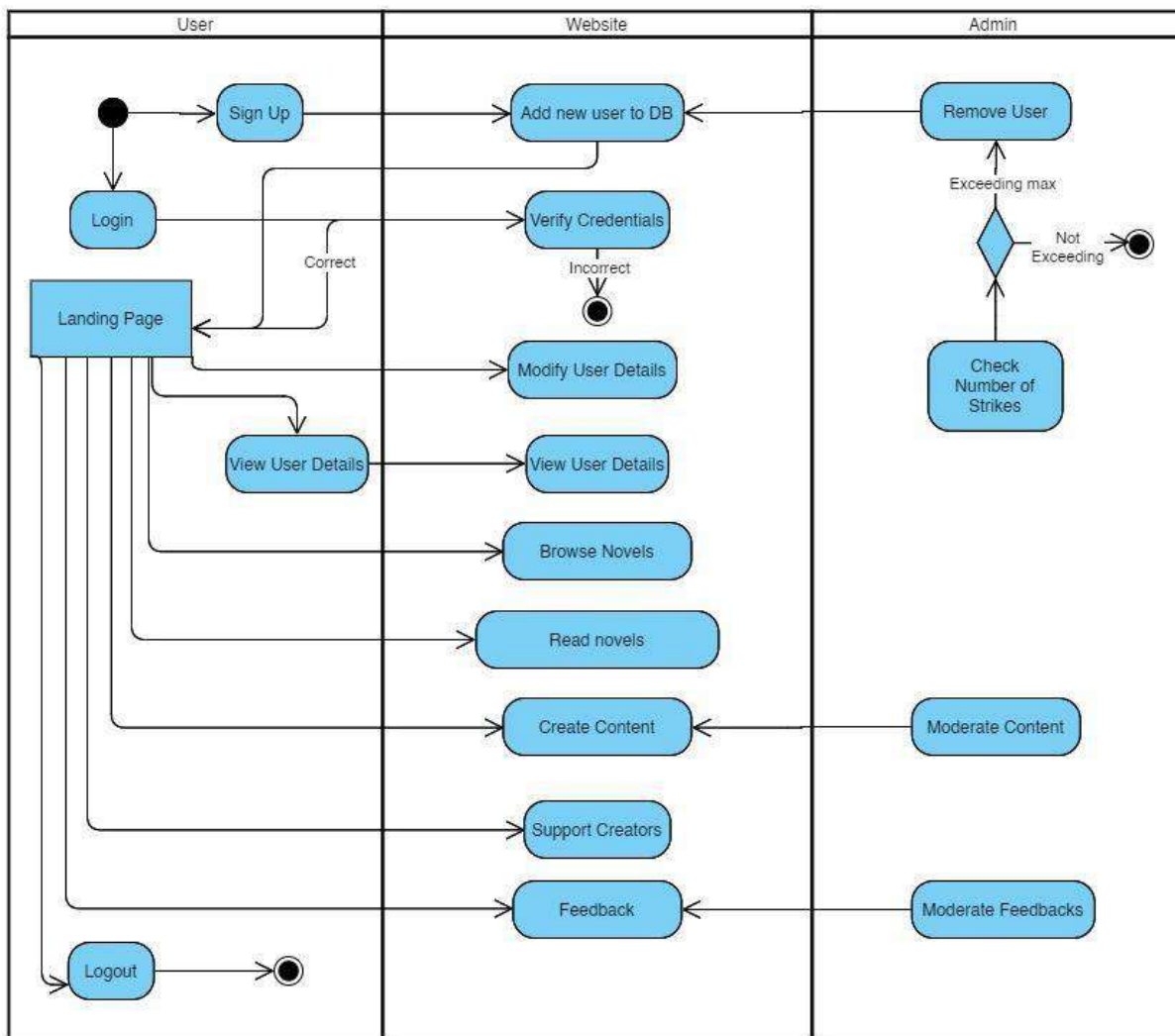


#### 4.4.3. UML diagram

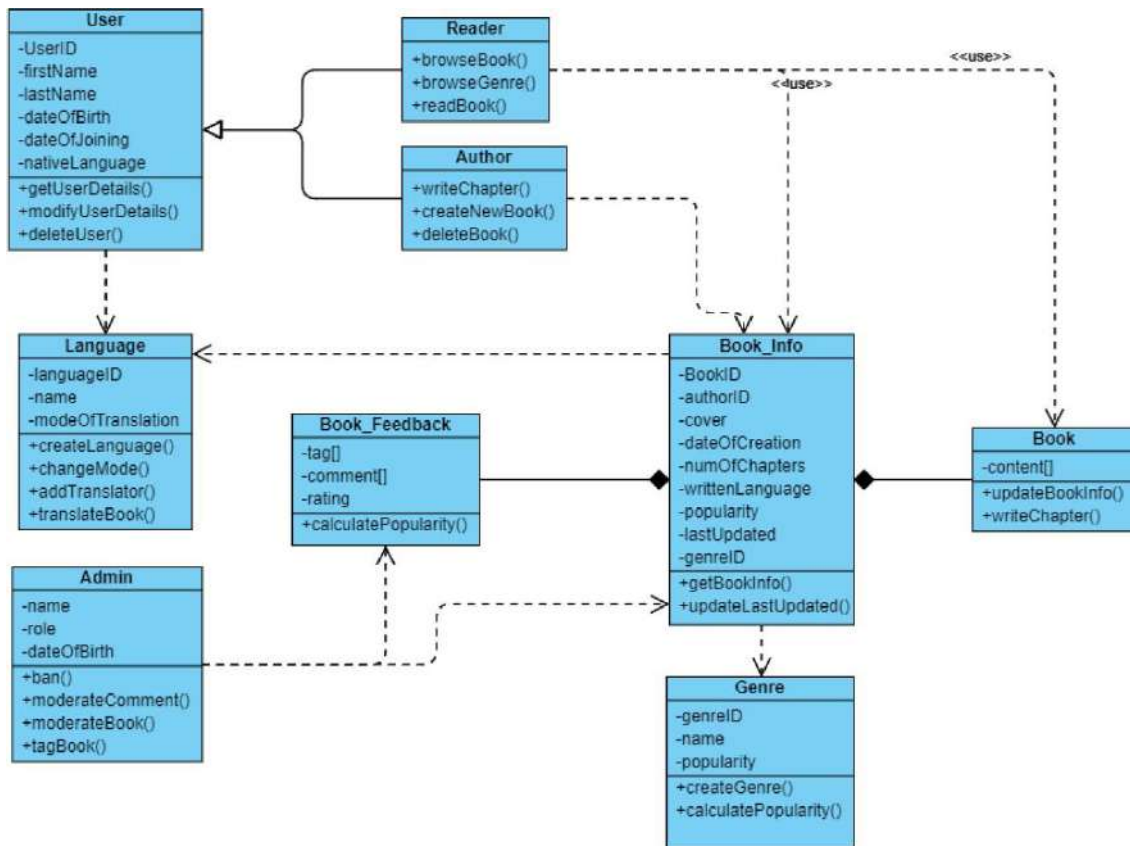
- **Use Case Diagram:**



• **Activity Diagram:**

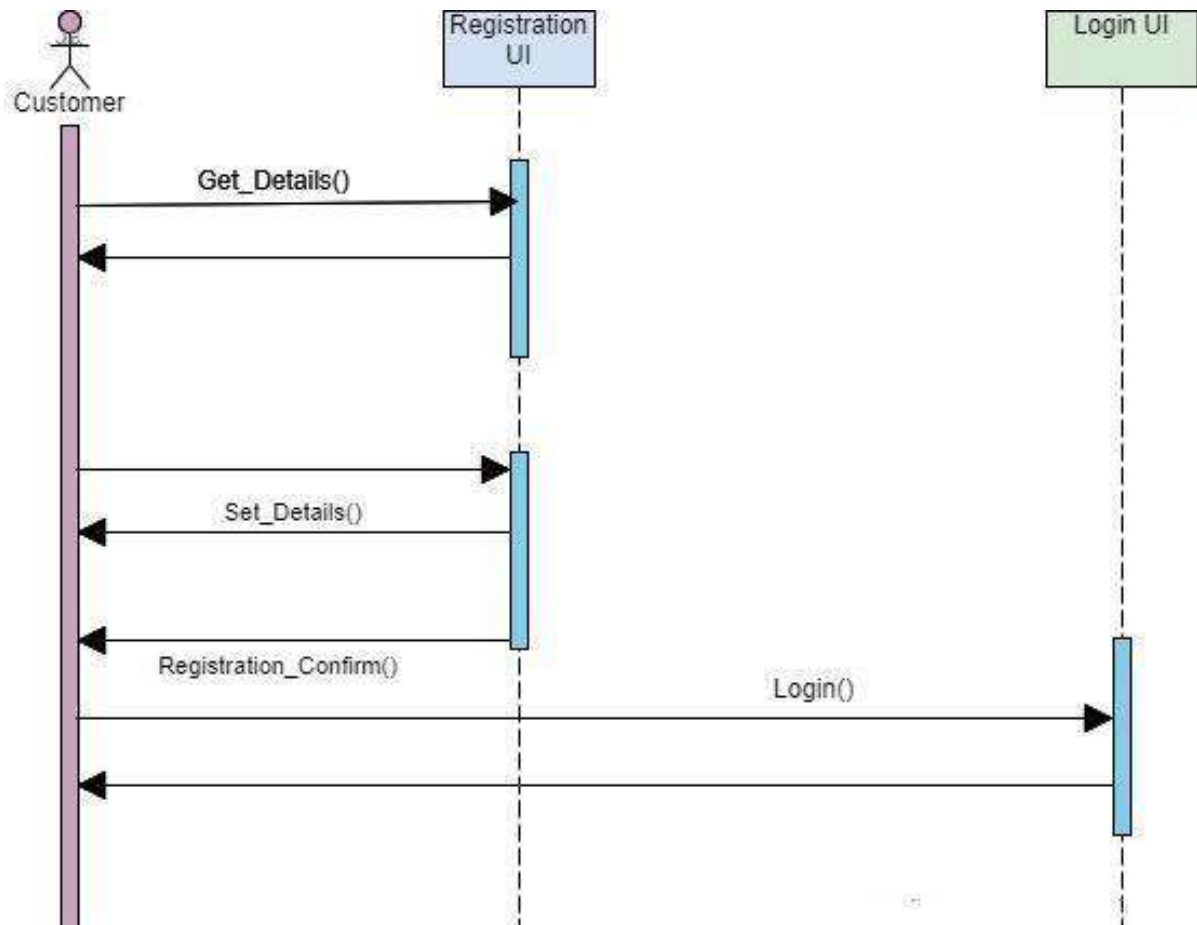


- **Class Diagram:**

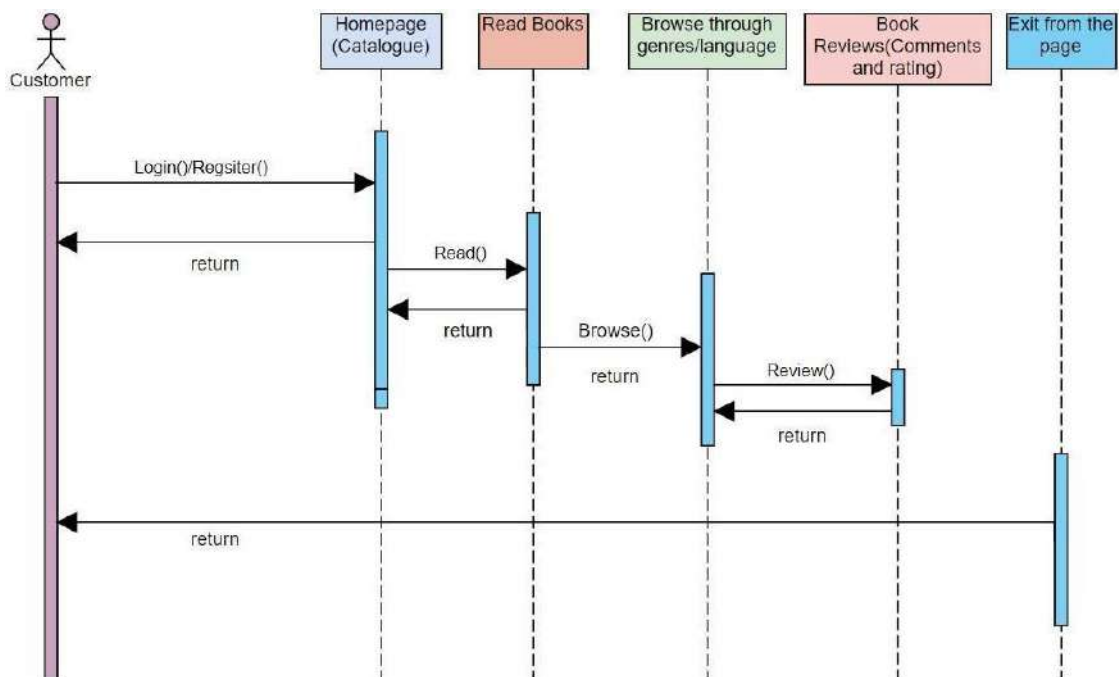


- **Sequence Diagram:**

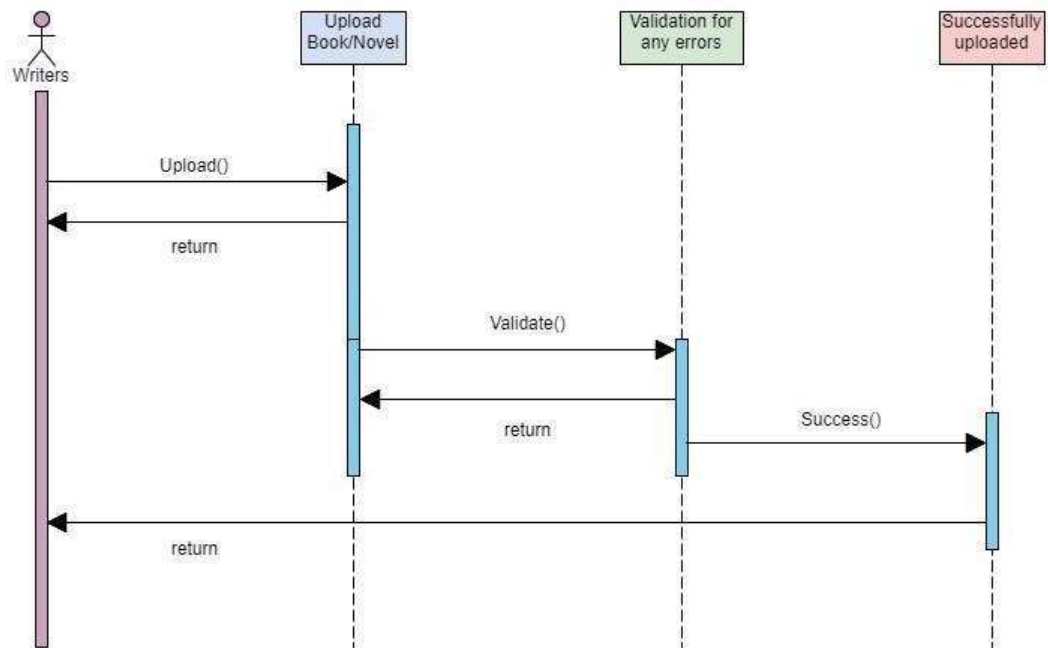
**A. For Login/Signup**



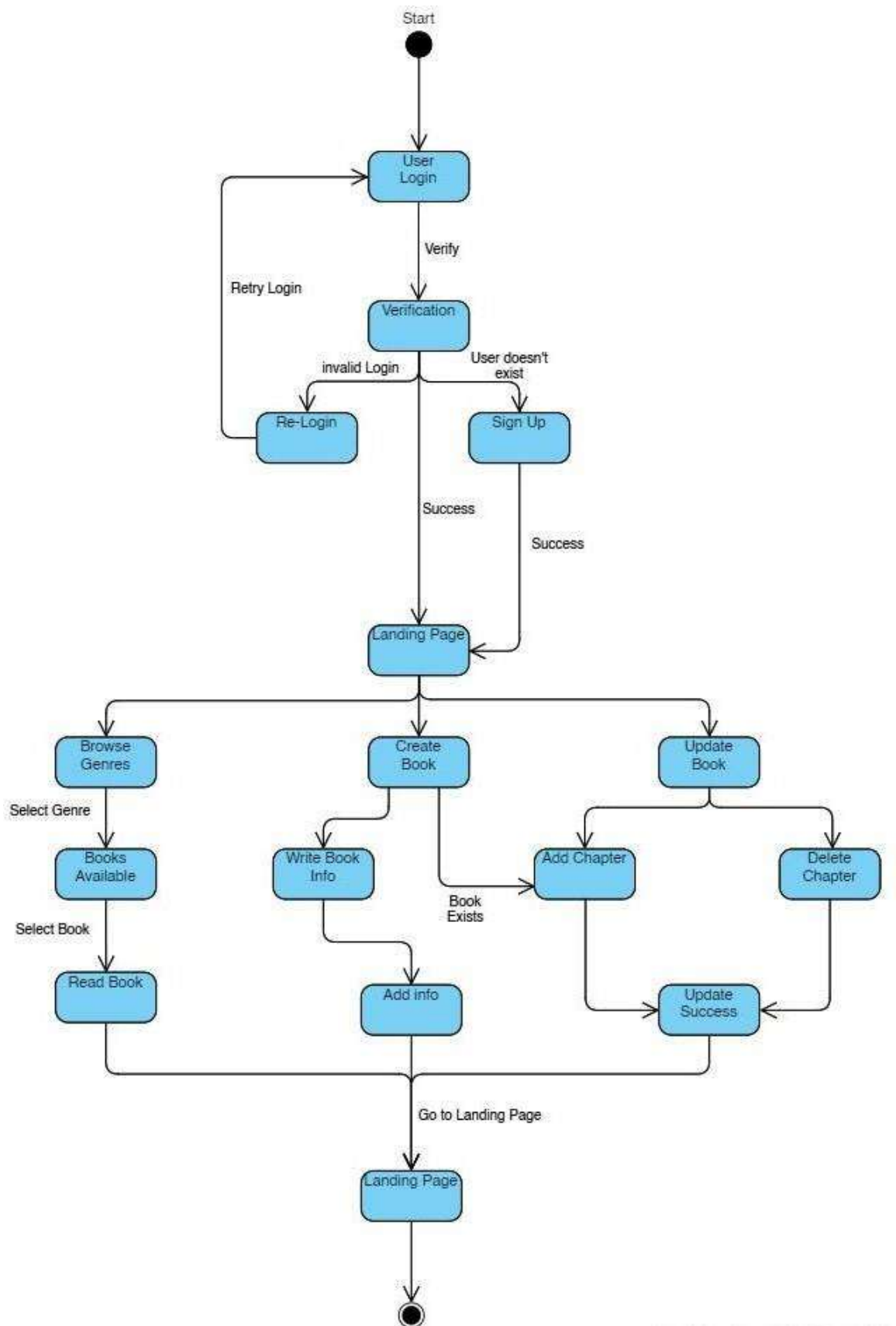
**B. For Users**



### c. For Writers



- State Diagram:



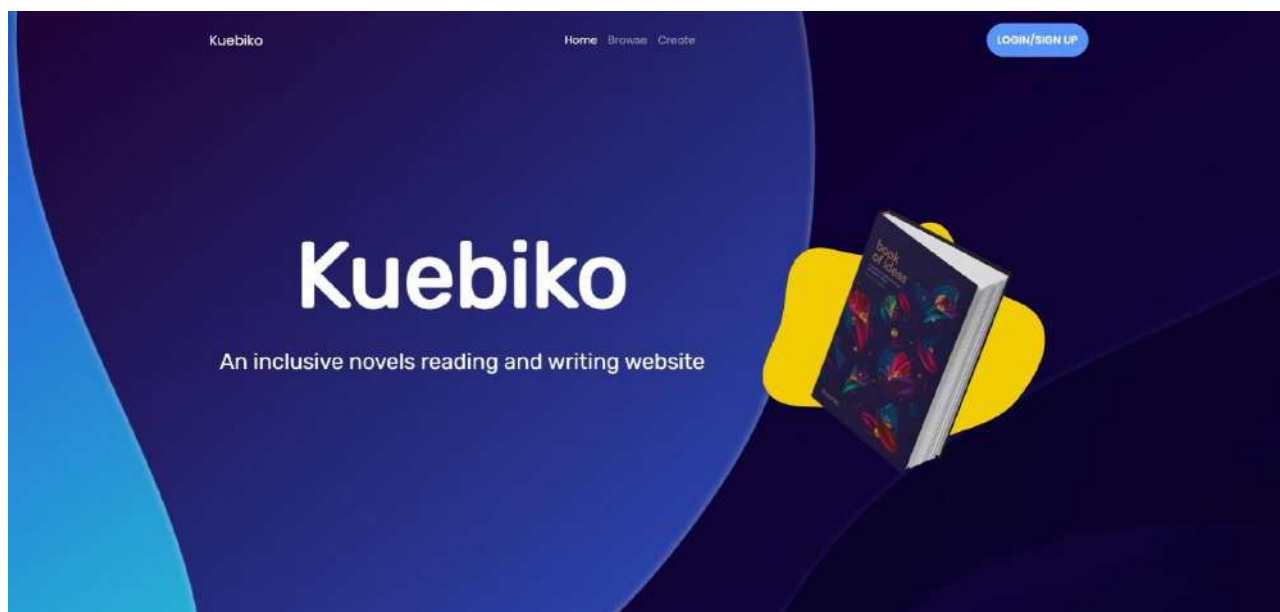


## 5. Implementation and Testing:

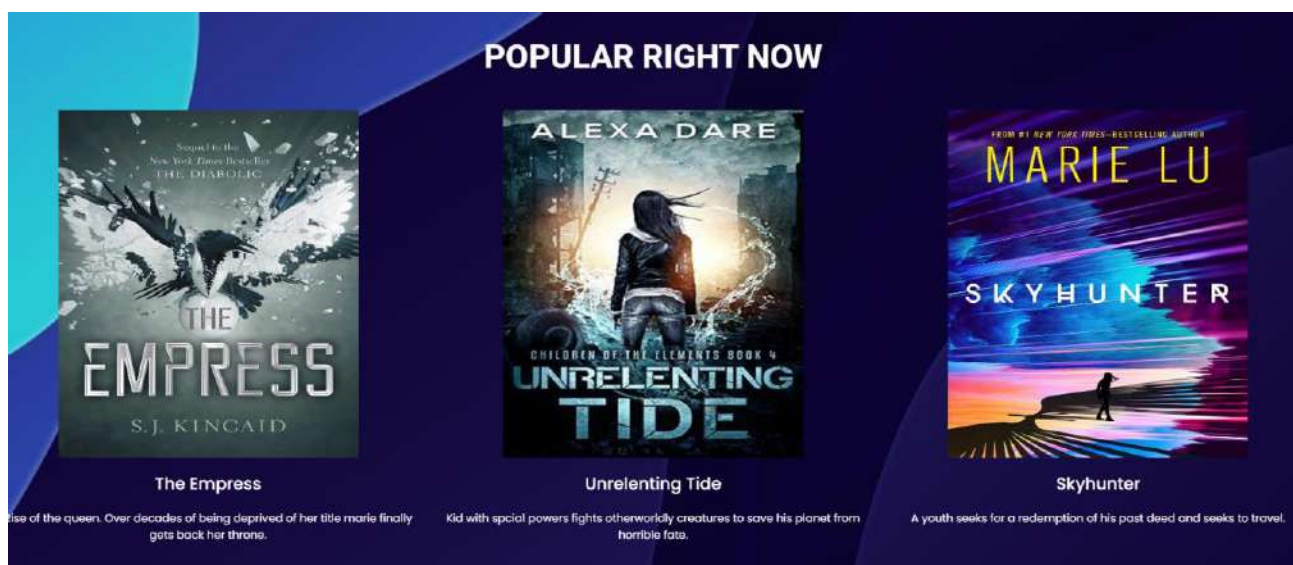
The full Frontend Code is available on <https://github.com/Padm0069/Kuebiko>. The full Backend Code is available on [https://github.com/Padm0069/Kuebiko\\_Backend](https://github.com/Padm0069/Kuebiko_Backend).

### 5.1. Implementation details

- **Landing Page**

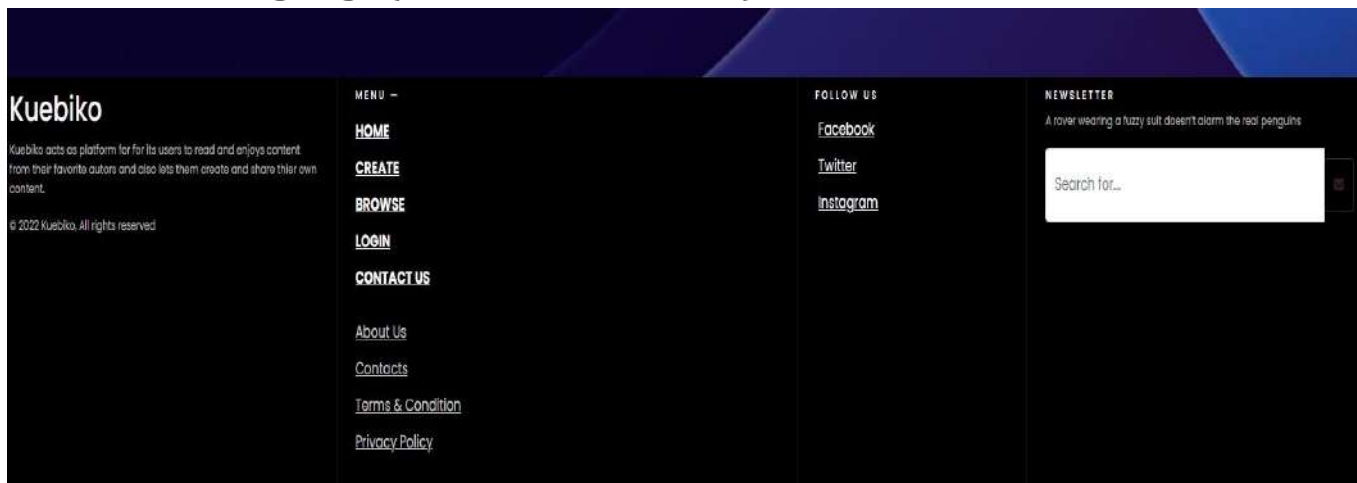


- **Landing Page (Popular Right Now Section)**

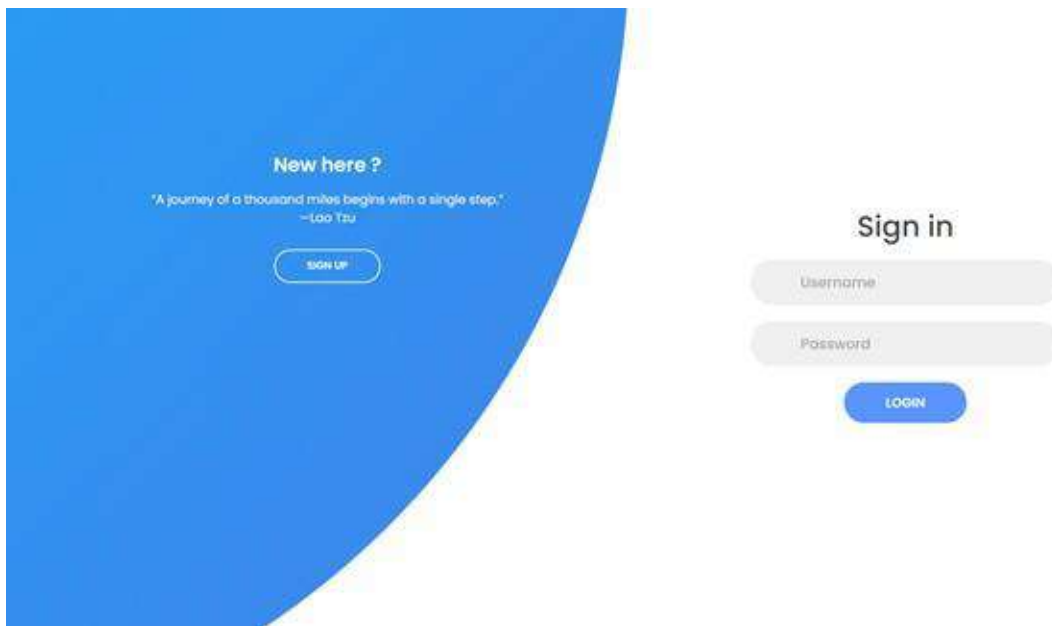


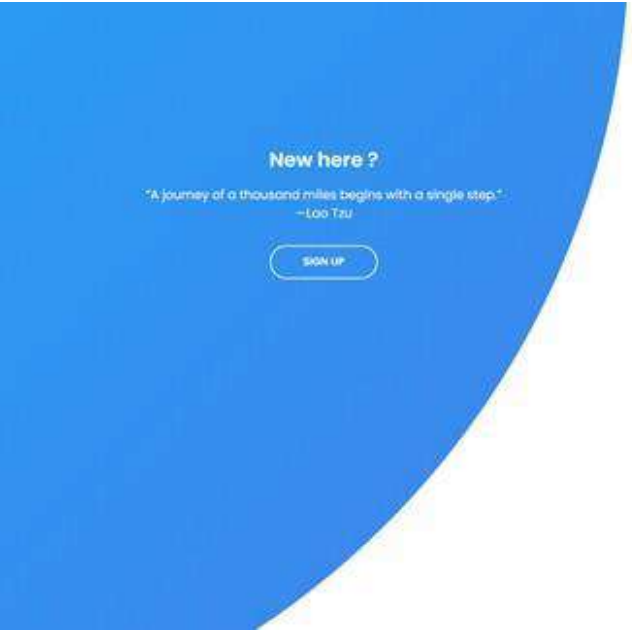


- **Landing Page (Features and Footer)**

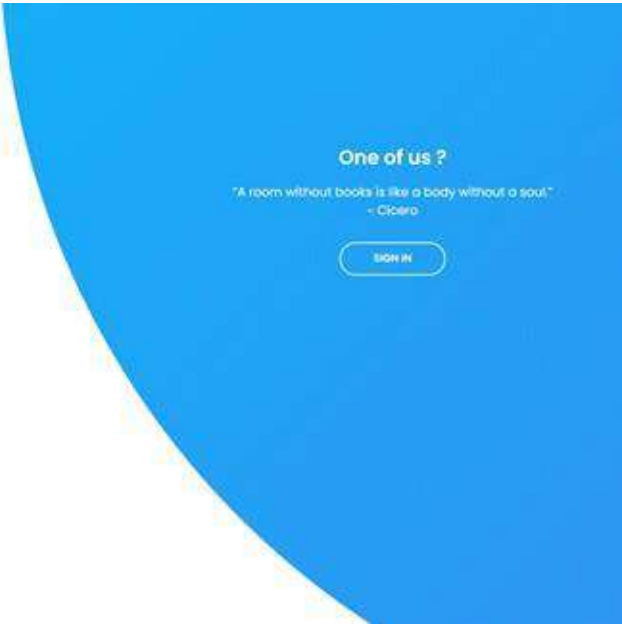


- **Sign In**



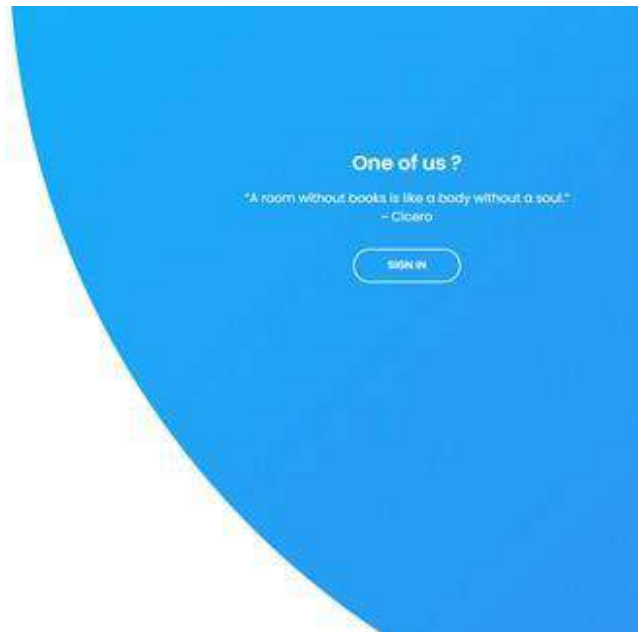


- Sign Up



### Sign up

SIGN UP



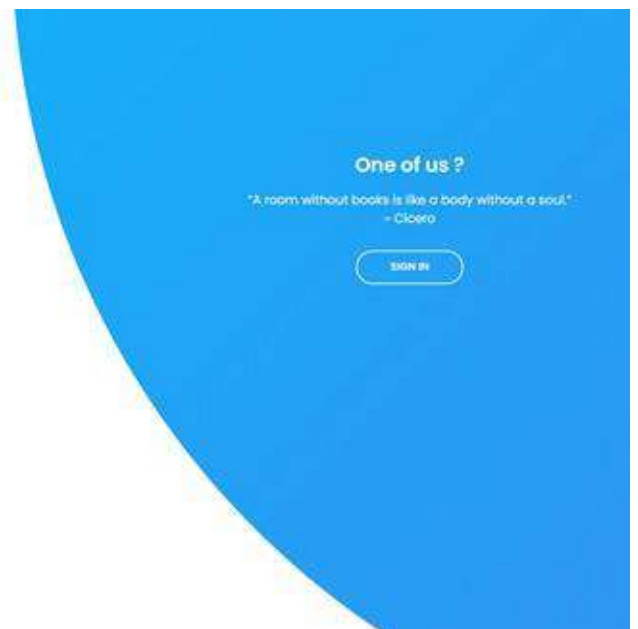
### Sign up

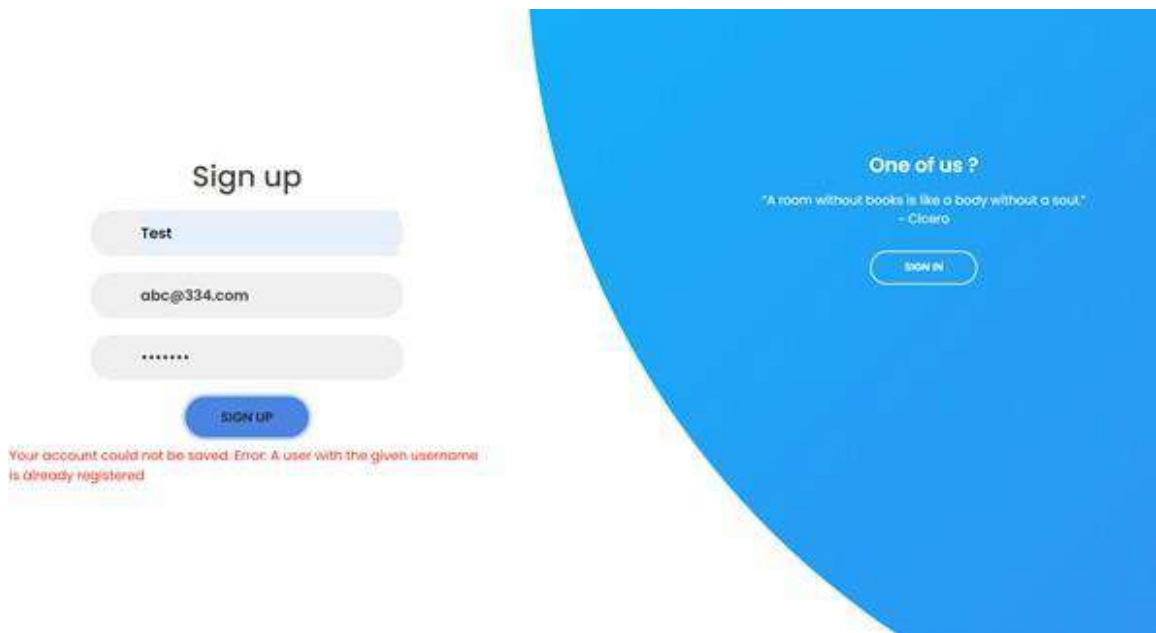
Test

abc

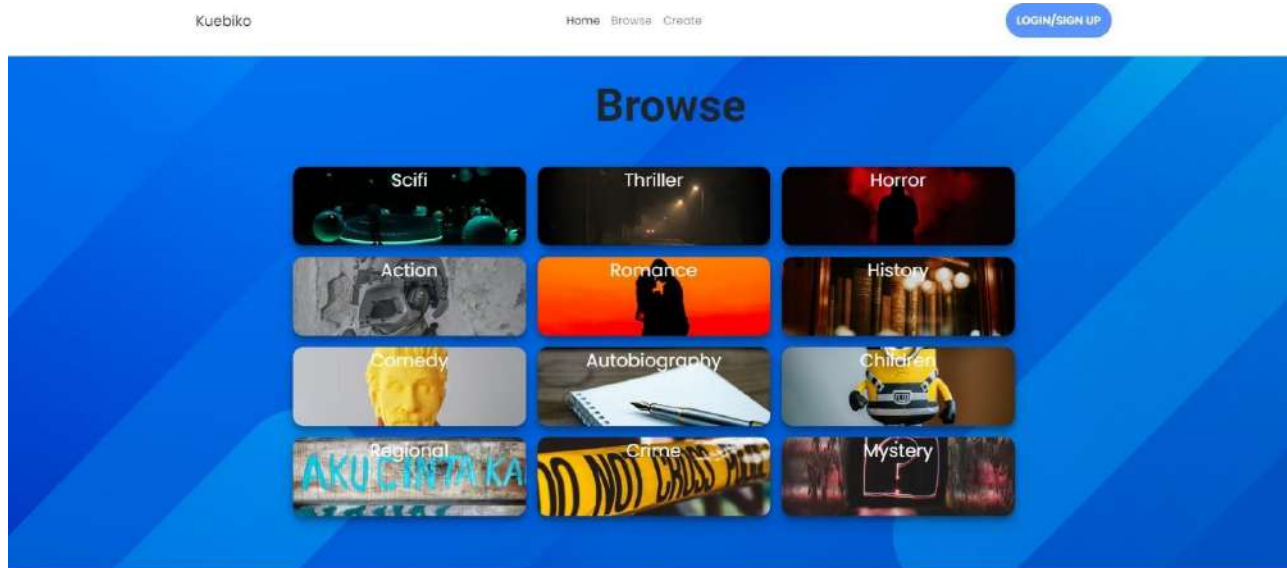
Please include an '@' in the email address. 'abc' is missing an '@'.

SIGN UP





- **Browse Page**



## ● Book Section

Kuebiko

Home Browse Create

LOGIN/SIGN UP



### Insignia

Mordred gets her revenge and gets back to her lover and they sail over to moon.

READ



### Crooked Kingdom

A dark and ominous creature rises from the depths of hell to take over the world

READ



### Game of Thrones

Tale of love and war that grows imminent over time. The tale of betrayal and survival

READ



## ● Reading Section

Kuebiko

Home Browse Create

LOGIN/SIGN UP



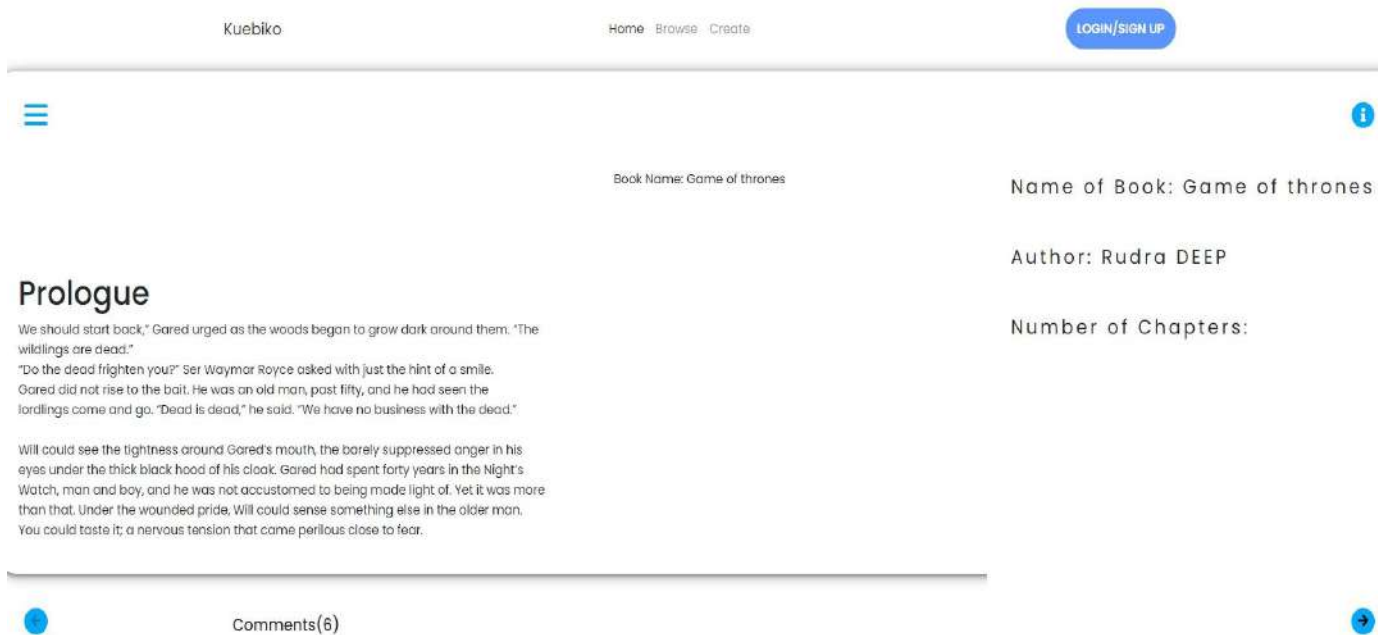
Book Name: Game of thrones

### Prologue

We should start back," Gared urged as the woods began to grow dark around them. "The wildlings are dead."

"Do the dead frighten you?" Ser Waymar Royce asked with just the hint of a smile. Gared did not rise to the bait. He was an old man, past fifty, and he had seen the lordlings come and go. "Dead is dead," he said. "We have no business with the dead."

Will could see the tightness around Gared's mouth, the barely suppressed anger in his eyes under the thick black hood of his cloak. Gared had spent forty years in the Night's Watch, man and boy, and he was not accustomed to being made light of. Yet it was more than that. Under the wounded pride, Will could sense something else in the older man. You could taste it; a nervous tension that came perilous close to fear.



## ● Book Creation

This screenshot shows the 'Create Book' form on the Kuebiko platform. The form is titled 'Create Book' and has four input fields: 'Name', 'Genre', 'Language', and a 'Create' button. The form is set against a light blue background.This screenshot shows the 'Create Book' form on the Kuebiko platform with sample data entered. The form is titled 'Create Book' and has four input fields: 'Name' (Insallah), 'Genre' (Romantic), 'Language' (Urdu), and a 'Create' button. A red message 'New Book Created' is displayed below the form. The form is set against a light blue background.



- **Chapter Creation**

Author:

Add Chapter

Book Name:

insallah

Chapter Name:

Context

Chapter Content:

I'll die before I'll know who I am

Create

success

● All Books

Kuebiko	Home	Browse	Create	LOGIN/SIGN UP
Game of thrones				
Insignia				
Crooked Kingdom				
Unrelenting Tide				
Skyhunter				
Game of Thrones				
Game of Thrones				
दायटियन				
Insallah				

## 5.2. Testing

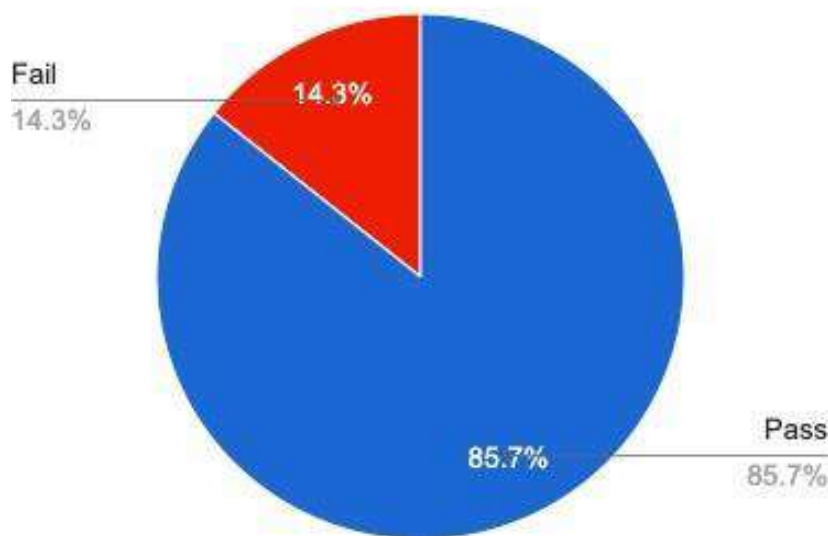
### 5.2.1. Test Case

Test Case Id	Activity	Inputs	Expected Result	Actual Result	Status	Comment
TC-01	Enter the valid username/email and password	Email Id and password	User successfully logs in	User successfully logs in and the user is redirected to the user dashboard	Pass	The log in is successful
TC-02	Enter the invalid username/email and password	Email Id and password	User is unable to log in because the entered data doesn't match any data in the database	User is unable to log in because the entered data doesn't match any data in the database and is shown the error "Username or Password is incorrect"	Fail	Invalid log in attempt was denied

T C — 0 3	Enter the name of the book	Name of the book	Boo k is fou nd	Boo k is fou nd	Pass	User finds the book they entered in the search bar
T C — 0 4	Create new book	Name= "exampl e" Langua ge= "English " Genre= "exampl e"	New book is created	New Book is created and user is navigated to user dashboar d	Pass	User successfu lly creates the new book and uploads it
T C — 0 5	Search for book	Languag e= "Englis h" Genre=	“Boo k is Foun d”	“Book is Found ” is shown	Pass	User entered data matches the

		"example"				data in the database
T C - 0 6	Language selection	Name="example" Genre="example"	Language is Found	Message "Language is Selected" is Shown	Pass	User finds the language they were looking for
T C - 0 7	Genre selection	Name="example" Language="English"	Genre is Found	"Error: Genre is Selected" is Shown	Pass	User finds the genre they were looking for

### Testing



We have used **QAWolf** to implement the automatic testing. It is a free open-source software that can be used on all browsers, supports various languages which includes JavaScript and so on which are the languages use by us in creating this project.

A good quality testing tool helps the tester to identify any critical bugs in the project. Without a proper tool, the manual process of going through the entire code to finding a bug, then noting it down and informing the developer takes a lot of time and often during retesting, the tester may miss some critical bug.

pass: todomvc

C:\Users\91987>npx qawolf test

Test: chromium

npx jest --config="node\_modules/qawolf/js-jest.config.json" --rootDir=.qawolf --testTimeout=60000

PASS .qawolf/todomvc.test.js (5.244 s)

FAIL .qawolf/qawolf.test.js (32.318 s)

• qawolf

Failed: "waitFor timed out after 30000ms"

```
15 | });
16 |
> 17 | test("qawolf", async () => {
    |   ^
18 |   const page = await qawolf.waitForPage(context, 0, { waitUntil: "domcontentloaded" });
19 |   await page.reload();
20 |   // 🦊 create code here
```

at Env.it (../node\_modules/jest-jasmine2/build/jasmineAsyncInstall.js:95:24)

at Object.<anonymous> (qawolf.test.js:17:1)

Test Suites: 1 failed, 1 passed, 2 total

Tests: 1 failed, 1 passed, 2 total

Snapshots: 0 total

Time: 33.403 s, estimated 112 s

Ran all test suites.

## 6. Conclusion, Limitations and Scope for future Work:

This project helped us improve our skills as well as get a platform easily accessible by people to share their views, thoughts and creativity. It will also act as a place that people will visit in their leisure time to dive into the ocean of literature, novels and much more. It also allows smaller authors to showcase their talent and gain more followers. With the implementation of text to speech feature users with visual impairments can use the website with ease. People can also use this while doing their other work in order to enjoy their books.

The current model of the website doesn't have any module that allows the user to interact with other users other than the comment section. Also there is no feature to show appreciation to the authors by contributing money to help them have a better livelihood.

We would like to implement a module which enables users to interact among themselves as the only interaction is through the comment section. We would also like to include a feature that lets users show appreciation to the authors by donating money. This will motivate the authors to write more and improve their content for better feedback.

## 7. References:

### 7.1. List of Websites

- <https://www.webnovel.com/>
- <https://www.audible.in/>
- <https://www.wattpad.com/>
- <https://kobatochan.com/>
- <https://www.npmjs.com/>
- <https://getbootstrap.com/docs/5.0/getting-started/introduction/>
- <https://reactjs.org/>
- <https://nodejs.org/>
- <https://mongoosejs.com/>
- <https://www.mongodb.com/>

### 7.2. List of Books

- I. Paul Deitel, Harvey Deitel, Abbey Deitel, Internet & World Wide Web – How to Program, 5th edition, Pearson Education, 2012.
- II. R. Elmasri S. B. Navathe, Fundamentals of Database Systems, Addison Wesley, 2015



# Kubeiko: An inclusive novels reading and writing website

*Written by*

**Anushka Srivastava**

**Padm Keshav**

**Rudhra Deep Biswas**

*Mentored by*

**Dr. SWARNALATHA P**

**Keywords:** E-Commerce, Web Development, Database, HTML, JavaScript, Novel Reader, Text-to-Speech, MVC Architecture, Figma,

## **Introduction:**

Undoubtedly, internet has changed the way we live our lives in today's fast paced world. Today everything from our daily necessities to every type of essential is available on internet. Everyone, be it old or young when interested in reading a book just click on their browser and look up the title and start reading it online. Kubeiko is designed to assist this process where such users can find all the books they wish to read at one place instead of going from website to website. The common problem that we noticed with the websites that are currently available is that most of their content is centred towards English language and native speakers and writers are not able to use it these websites to the fullest. These platforms are taking advantage of the lack of competition and set the prices for viewing the books or novels higher than necessary which makes the user doubtful whether it's worth spending the money or not. The website has modules which assist people to translate an novel into their own desired language through Google translate, or if there are enough requests, translators can take up a novel and translate them to the requested language.

## **Literature Survey:**

Author and Year	Title	Concept	Implementation	Dataset	Limitations/ Future Research
-----------------	-------	---------	----------------	---------	---------------------------------

MOHAMMED A. ABDALA And NOOR AHMED KHIDER	ONLINE E-BOOK STORE WEBSITE DESIGN	The system was designed based on study of famous online book stores such as "Amazon" or "Google books" with the Iraqi market environment in mind.	The website logo is displayed in the top of the page, moreover, links to home page, search, free section and shopping cart below this logo. On the left side of the page there is a list of the different book genres or categories that the user can choose from. The content of each page are displayed in the middle area of the page.	Books Database, Bank Database	Problem with the currently available websites is that most of them are centred towards content in English language and most viewers and writers are not able to use it to the fullest.
--	------------------------------------	---	---	-------------------------------	--

## Methodology:

Project Kuebiko is a new, improved and fresh attempt to design a product which is intuitive, inclusive and easy to traverse and use.

JavaScript and its frameworks were used for the development of this website. It has been designed as a standalone software that can run on any device which has a web browser.

All the interfaces of the website follow a consistent theme and clear structure which is appealing to the user. The design structure for the website is designed using FIGMA. The prototype application was modelled consisting of all the features and functionalities the website has to offer to the users.

Various features such as book searching, adding new novels, text-to-speech features, etc. are added using the JavaScript framework to ensure customer satisfaction.

The text-to-speech was implemented using a JavaScript npm text-to-speech package. The particular feature was added keeping in mind the ease of usability for readers who are visually impaired.

The application will be able to run on any device, since the hardware interfaces for this system would include Desktop interfaces containing Monitor, Keyboard and Mouse and smartphones and tablets with Touch screens and Virtual Keyboards. The hardware interface for the user would be any PC having a configuration of P-IV and above 2GB HDD for loading any OS so that it can interact with the system without any problem.

Kuebiko would take help of a database for storing and managing the novels being uploaded on the website by the authors. So, an access to the database management system is required. When

such an event occurs, the system establishes connection to the database management system, once the connection is created; the client program can communicate with the database management system. The website was linked with a data set created by the authors which will be regularly updated. The dataset consists of novel copies, the author's information, user's information and so on.

Furthermore, a security check algorithm will be implemented to avoid the users from putting up malicious or inappropriate content. System will use secured database. Normal users can just read information but they cannot edit or modify anything except their personal and some other information. System will have different types of users and every user has access constraints. Proper user authentication should be provided. No one should be able to hack users' password. There should be separate accounts for admin and members such that no member can access the database and only admin has the rights to update the database. The users should keep their system access credentials to themselves in order to prevent unauthorized access and misuse of data. If at any point a user decides to discontinue their services, the system should allow them to do so after proper authorization and proof. In case the user forgets/ misplaces the key to login into their account they should be provided with the facility to recover their account or in the worst case if the account is not recoverable, terminate the account.

The database may get crashed at any certain time due to virus or operating system failure. Therefore, it is required to take the database backup so that the database is not lost. Proper UPS/inverter facility should be there in case of power supply failure.

A real time review system algorithm will also be implemented which will allow users reading the same content to interact with each other in real time.

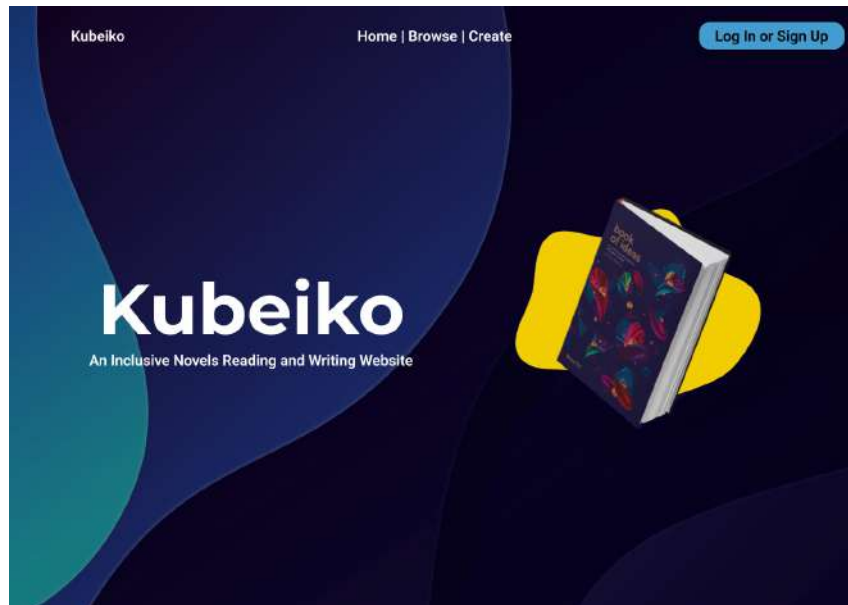
For communications, the TCP protocol shall be used. A client program creates a socket on its end of the communication and attempts to connect that socket to a server. When the connection is made, the server creates a socket object on its end of the communication. The client and server can now communicate by writing to and reading from the socket.

Express will be used for backend API generation. The frontend framework will use an asynchronous function to communicate with the API endpoints. The backend will also communicate with the Database with the help of the API of the hosting site.

MVC (Model-View-Controller) Architectural Style is implemented taking into consideration that there are multiple ways to view and interact with data. MVC is use to avoid the problem where due to intermixing of the code between the process logic and interface, the modularity of the application is reduced and it becomes difficult to maintain. The main advantage of the used style is that it allows the data to change independent of its representation and vice versa.

## Results and Discussion:

- *Landing Page*



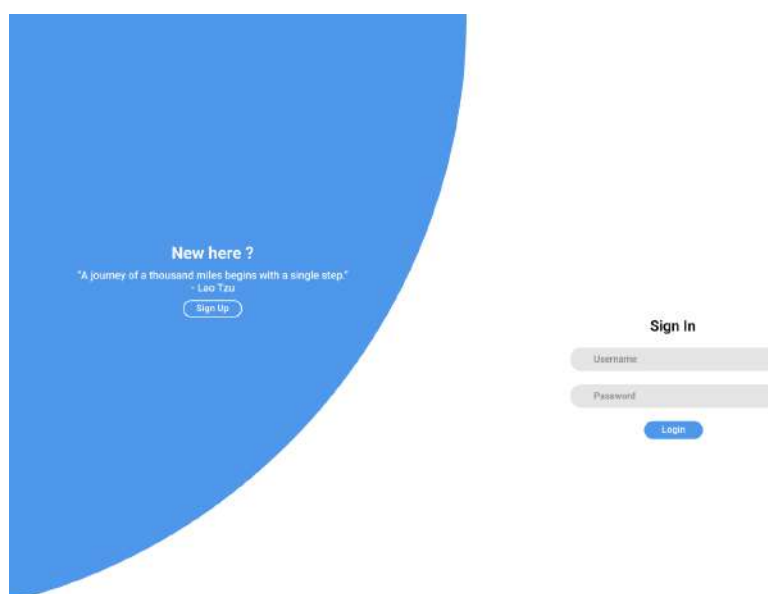
**Description:** This is the landing page that the User sees when they load the website.

**Input :** Toggle options depending on the current situation of the User (Home, Browse or Create)

**Output:** Redirects the User to the page they selected

**Processing:** Redirects them to the selected option.

- *Log In Page*



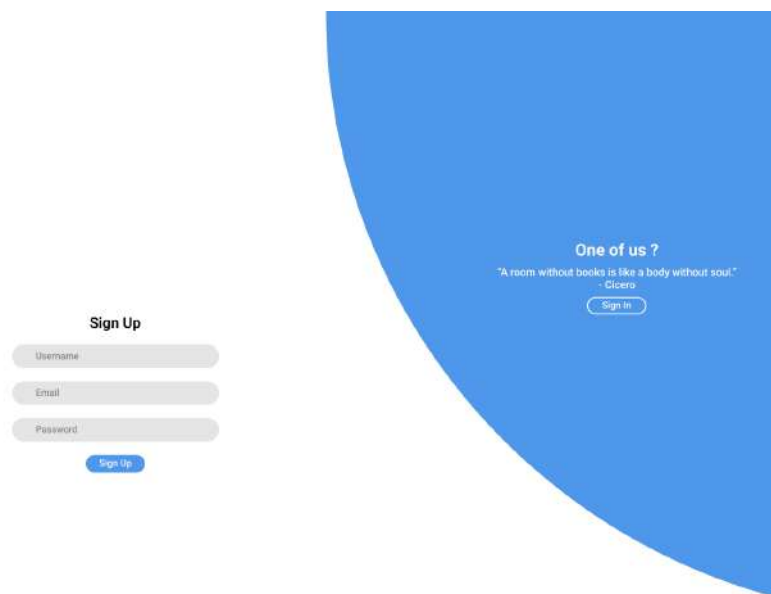
**Description:** This allows the User to save their progress, their search history and allows them to rate and comment on any book.

**Input:** Username and Password

**Output:** Logged in if credentials are correct; Error screen if incorrect

**Processing:** Search Username in database. If none found redirect to sign up. If found, check password. If matched, successfully log in User.

- *Sign Up Page*



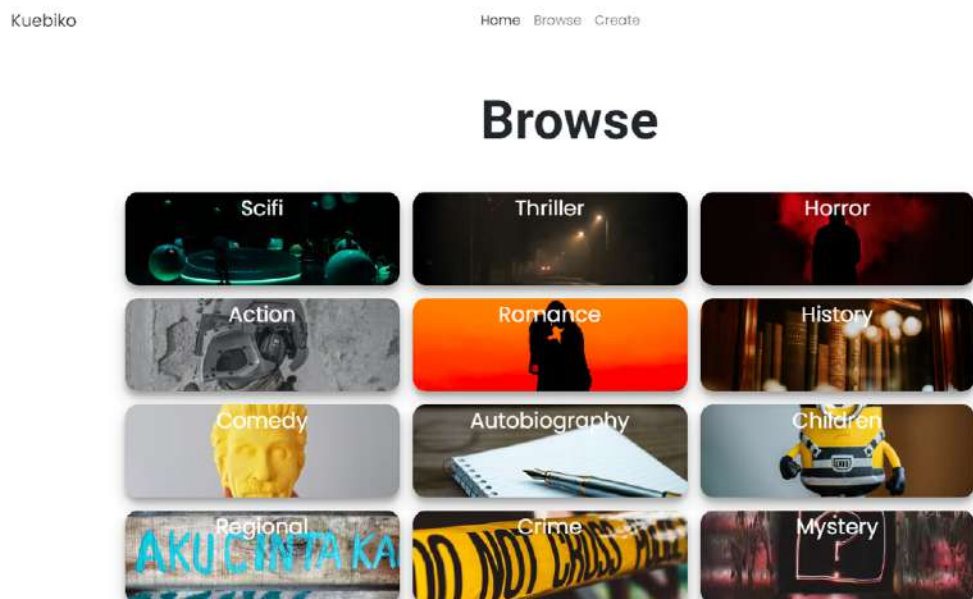
**Description:** This allows the User to create their own account on the site.

**Input:** User details and Password

**Output:** Add user to database. Display Sign Up successful.

**Processing:** Add User details according to schema to User Database.

- *Browse Books Page*



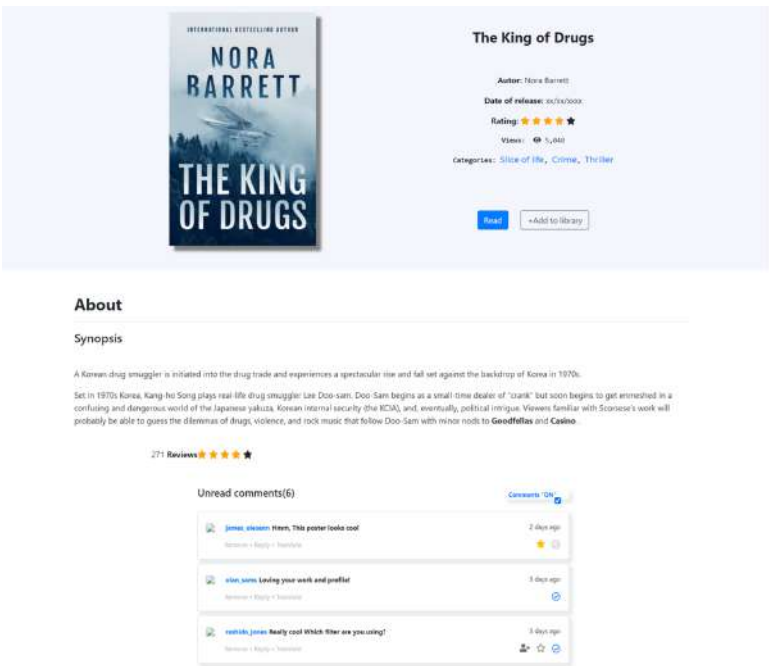
**Description:** This allows a user to search a particular book or author by keywords and find them. If multiple results are found, the user can search according to specific filters.

**Input:** Keywords

**Output:** List of Books, Authors, Tags or Genres that match the keywords

**Processing:** Search keywords in the Book database. If keywords match Book name, Author name, Tag name or genre name, add to list. Finally display it.

- *View Contents of the Chapter Page*



**Description:** This allows the user to get a brief idea on the number of chapters present in the novel. They can get a summary of what happens in that chapter before they dive into the actual content.

**Input:** Chapter number of the Book that was clicked

**Output:** Page that contains chapter details.

**Processing:** Search database for chapter number, find it and display it.

- *Create a New Book Page*

The screenshot shows the 'Create Book' form on the Kuebiko website. The form is titled 'Create Book' and has three input fields: 'Name', 'Genre', and 'Language'. Each field is represented by a blue horizontal bar. Below these fields is a blue button labeled 'Create'. The website's header shows 'Kuebiko' on the left and 'Home Browse Create' on the right.

**Description:** Any user that wants to write/ submit a book can select this option. The user is led to a page where they can give their input and submit, after which it is published on the website.

**Input:** Details of the new book.

**Output:** Display new book created.

**Processing:** Input new book data into the database.

- *Add a New Chapter Page*

The screenshot shows the 'Add Chapter' form on the Kuebiko website. The form is titled 'Author: Add Chapter' and has three input fields: 'Book Name', 'Chapter Name', and 'Chapter Content'. Each field is represented by a blue horizontal bar. Below these fields is a blue button labeled 'Create'. The form is enclosed in a light blue border.

**Description:** This allows the user to add a new chapter to an already existing book.

**Precondition:** User has created a book.

**Input:** Heading and content of chapter.

**Output:** Display Chapter added and add chapter to database.

**Processing:** Checks for BookID of the selected book and adds the chapter to its Content module.

## Conclusion:

Problem with the currently available websites is that most of them are centred towards content in English language and native viewers and writers are not able to use it to the fullest.

The platforms are taking advantage of the lack of competition and the prices set for viewing the books or novels are high which make the user doubtful whether they should buy it.

The website has modules which assist people to translate an novel into their own desired language through Google translate, or if there are enough requests, translators can take up a novel and translate them to the requested language.

The plan is to provide free novels to the users and remuneration through ad revenue. Authors also have the freedom to choose a payment-based method for remuneration.

## References:

- <https://www.researchgate.net/publication/345653981> The Complete E-Commerce Book Design Build Maintain a Successful Web-based Business
- <https://www.researchgate.net/publication/260907838> Online E-Book Store Website Design
- [https://www.matec-conferences.org/articles/mateconf/pdf/2017/14/mateconf\\_gcmm2017\\_02045.pdf](https://www.matec-conferences.org/articles/mateconf/pdf/2017/14/mateconf_gcmm2017_02045.pdf)
- <https://thenewstack.io/the-open-book-project-an-ebook-reader-you-can-build-yourself/>
- Fundamental of Software Engineering By Rajiv Mall
- IEEE. IEEE Std 830-1998 IEEE Recommended Practice for Software Requirements Specifications. IEEE Computer Society, 1998.
- [www.wikipedia.org](http://www.wikipedia.org)

## Appendix A: Glossary

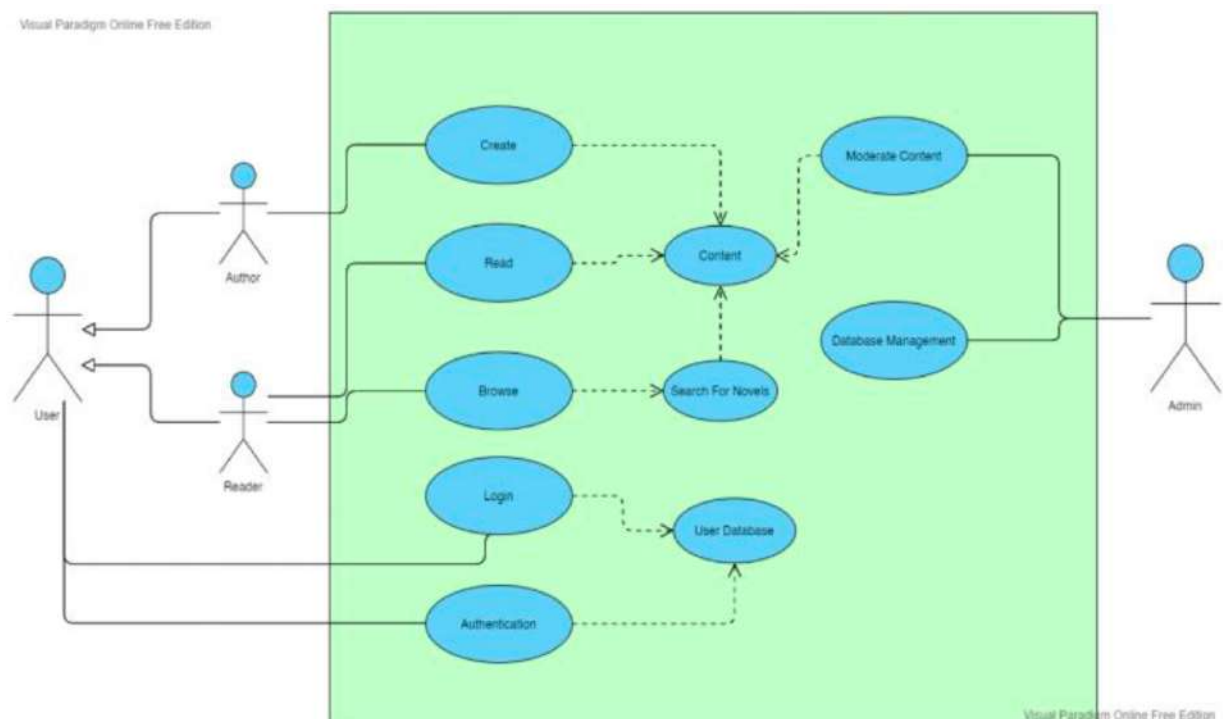
- *API (Application Programming Interface)* – A software intermediary that allows two applications to talk to each other.
- *Backend* - The server-side of the website users cannot see and interact with. It stores and arranges data, and also makes sure everything on the client-side of the website works fine.
- *Client* – A piece of computer hardware or software that accesses a service made available by a server as part of the client–server model of computer networks.



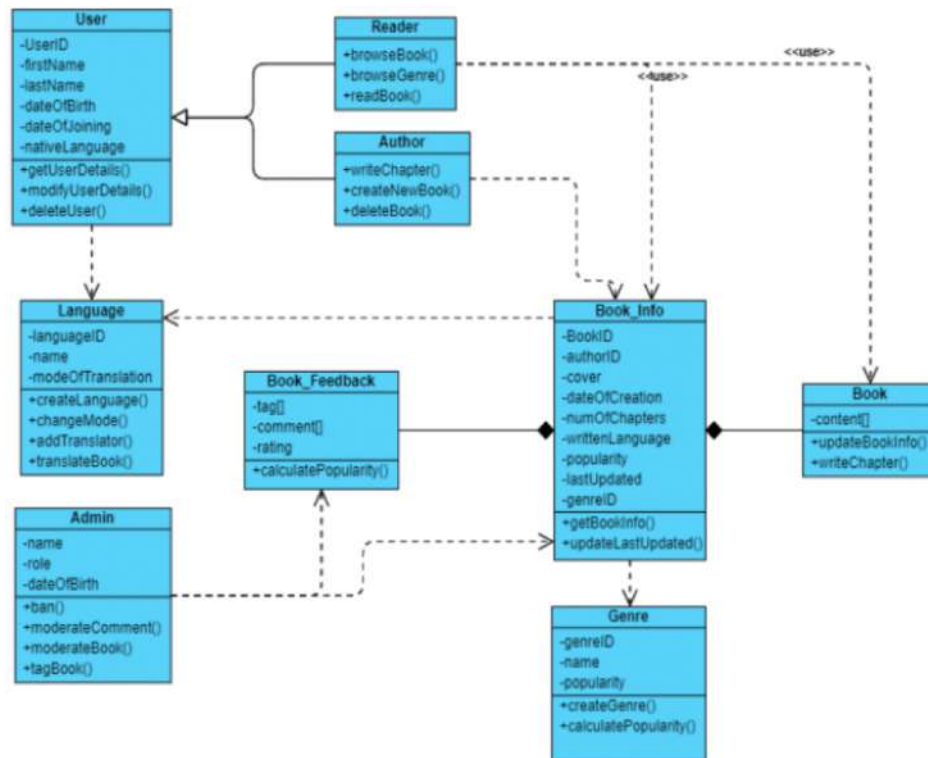
- **Database** – An organized collection of structured information, or data, typically stored electronically in a computer system.
- **Frontend** – The part of the website users can see and interact with such as the graphical user interface (GUI) and the command line including the design, navigating menus, texts, images, videos, etc.
- **Hardware** – Physical parts computer that cause processing of data.
- **HDD (Hard Disk Drives)** – An electro-mechanical data storage device that stores and retrieves digital data using magnetic storage and one or more rigid rapidly rotating platters coated with magnetic material.
- **JavaScript** – The programming language of the Web.
- **Protocols** – An established set of rules that determine how data is transmitted between different devices in the same network.
- **Interface** – A device or program enabling a user to communicate with a computer.
- **Software** – A set of instruction that tells a computer exactly what to do.
- **SRS** – It is the complete specification and description of requirements of software that needs to be fulfilled for successful development of software system.
- **Text to Speech** – Technology that enables text to be converted into speech sounds imitative of the human voice.
- **UI (User Interface)** – The series of screens, pages, and visual elements—like buttons and icons—that enable a person to interact with a product or service.

## Appendix B:

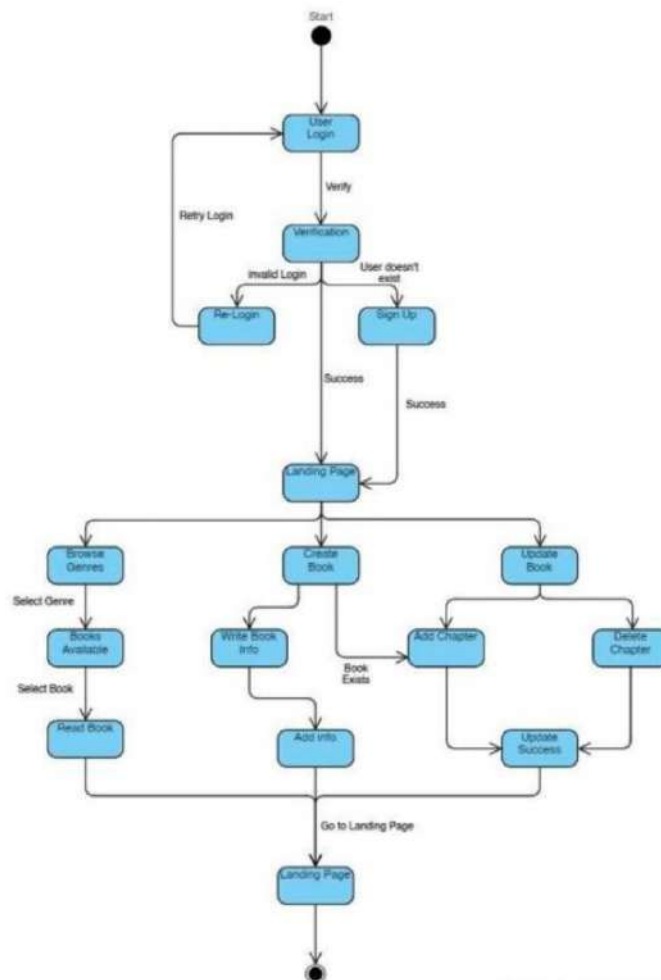
- **Use Case Diagram**



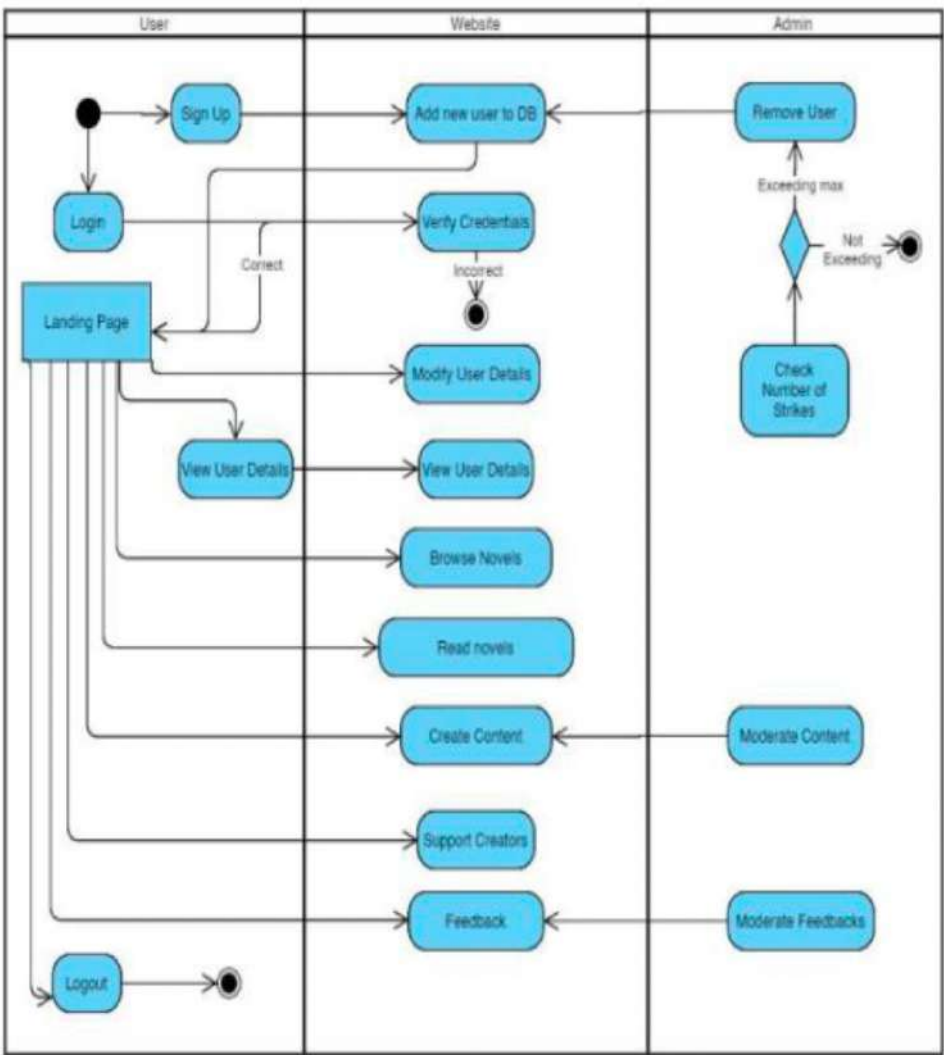
- *Class Diagram*



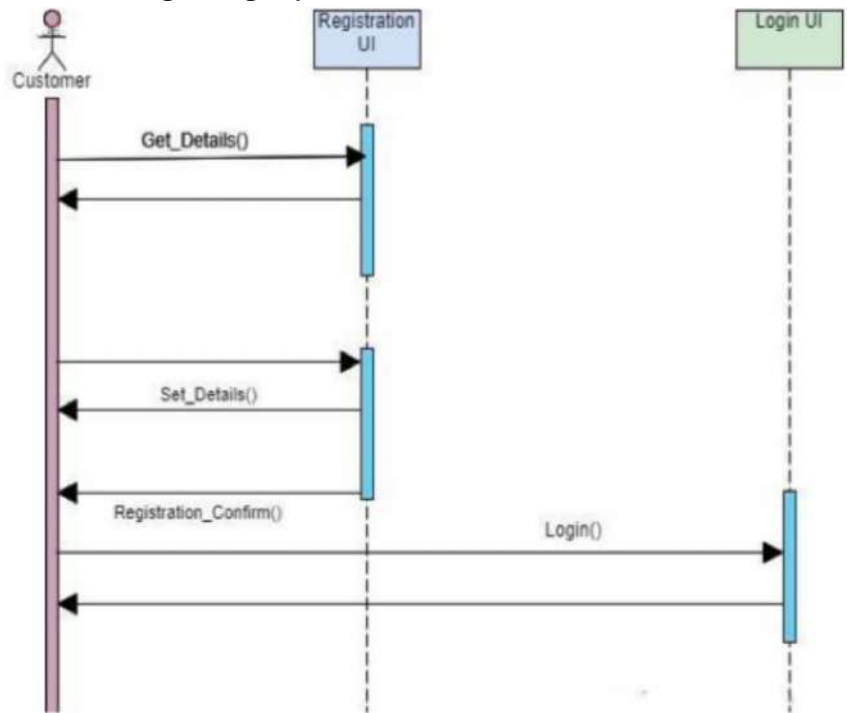
- *State Chart Diagram*



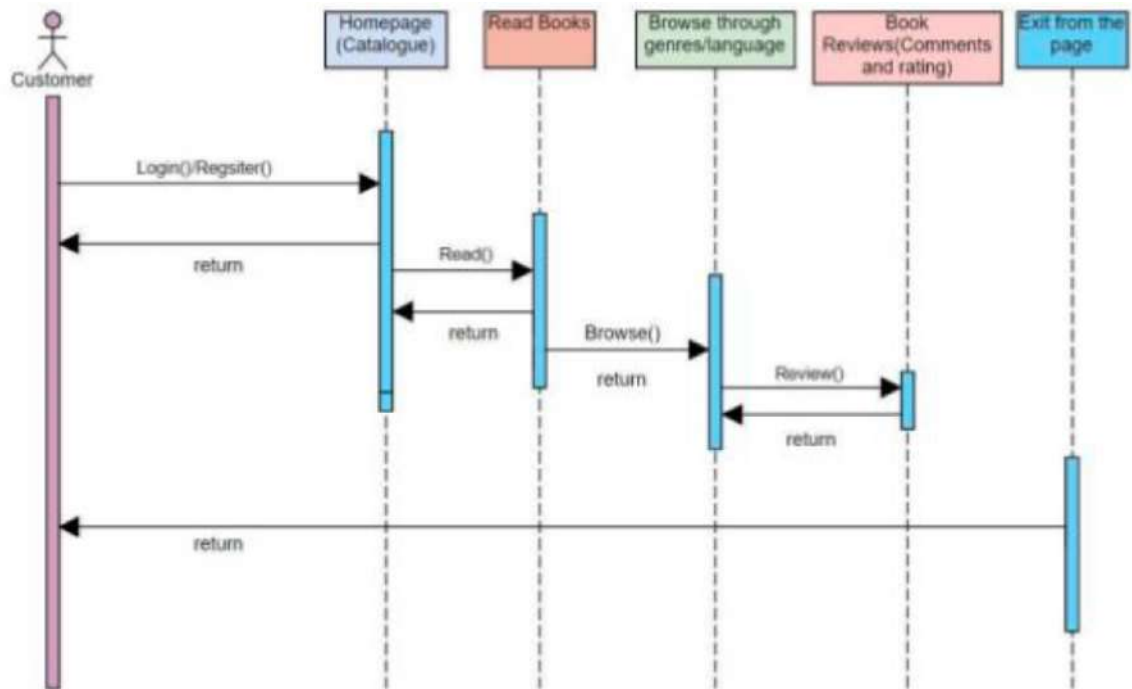
- Activity Diagram



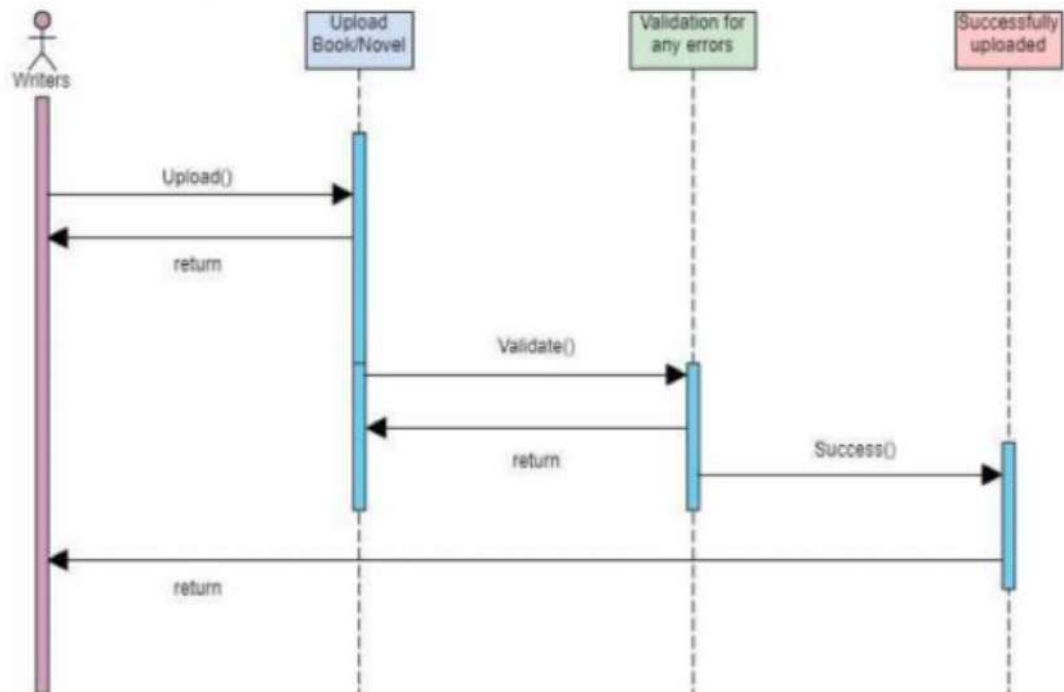
- Sequence Diagram: For Login/Signup



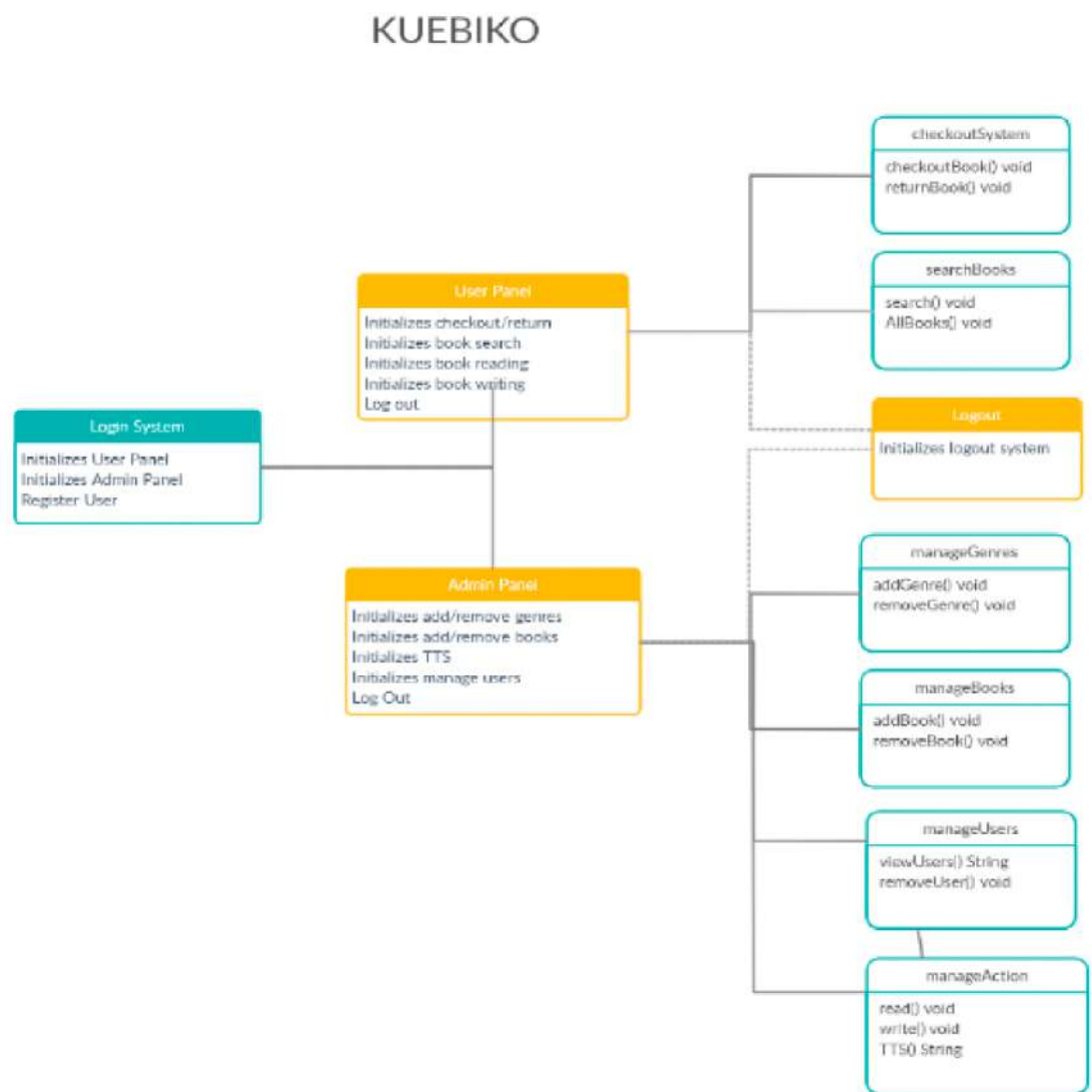
- *Sequence Diagram: For Users*



- *Sequence Diagram: For Writers*



- Deployment Diagram



- Collaboration Diagram



**Biography:**

**Anushka Srivastava:** I am a second year B. Tech student pursuing Computer Science and Engineering at Vellore Institute of Technology, Vellore. I have an inherent and deep interest in high-level programming languages such as C++, Java, Python, JavaScript, etc. I have a cognitive and analytical mindset. I possess knowledge and experience in the fields of Artificial Intelligence and Machine Learning.

**Padm Keshav:** A college junior seeking and learning various fields of tech, a regular challenger on CodeChef coding challenges and possessing a basic knowledge of front-end development with a interest of learning new frameworks and fields like blockchain.

**Rudhra Deep Biswas:** A college Second Year Cs Student seeking and learning various fields of tech, Active on Competitive Coding sites like CodeChef (3 stars), Leetcode (125+ problems solved), and possessing basic knowledge of Full-stack Web Development including react.js and JavaScript(ES6) basics with an interest in learning new frameworks and Technologies Constantly.



# Plagiarism Checker X Originality Report

**Similarity Found: 9%**

Date: Thursday, April 28, 2022

Statistics: 13 words Plagiarized / 150 Total words

Remarks: Low Plagiarism Detected - Your Document needs Optional Improvement.

---

Kubeiko: An inclusive novels reading and writing website **Written by Anushka Srivastava**  
Padm Keshav Rudhra Deep Biswas Mentored by Dr. SWARNALATHA P Keywords:  
E-Commerce, Web Development, Database, HTML, Java Script, Novel Reader,  
Text-to-Speech, MVC Architecture, Figma, Introduction: Undoubtedly, internet has  
changed the way we live our lives in today's fast paced world. Today everything from  
our daily necessities to every type of essential is available on internet. Everyone, be it old  
or young when interested in reading a book just click on their browser and look up the  
title and start reading it online.

Kubeiko **is designed to assist** this process where such users can find all the books they  
wish to read at one place instead of going from website to website. The common  
problem that we noticed with the websites that are currently available is that most of  
their content is centred towards English language and native speakers and writers are  
not able to use it these websites to the fullest. These platforms are ta

## INTERNET SOURCES:

---

3% - <https://yourstory.com/author/anushkasrivastava>

2% -

<https://amplitudedigital.com/in-todays-fast-paced-world-some-things-remain-timeless-classics/>

2% -

<https://www.quesba.com/questions/statement-cash-flows-designed-assist-users-assessing-following-except-569959>