Review 3

Group Members:

Gaurav Kumar Singh 19BCE2119

Piyush Rajput 19BCE0689

Peer-to-Peer Book Rental Service

Abstract

The project entitles "Peer-to-Peer Book Rental Portal" is a very effective, feasible online portal that facilitates the readers to read the books and magazines on rent reliably. If a reader wants to read a book, he/she has to purchase the book, which is not worth it for most users, especially for those who want to refer to only a part of the specific book, or does not want to keep the book after purchase or they can go to the library to lend a book where firstly they have to be a member and have to follow the library rules and regulations - like to return the book at the specified date that is given by the library. There is not much flexibility with the library management system and many places do not even have libraries and many of them are not very well maintained. So, a lot of time and money gets wasted on these. But with our application, the situation becomes very easy. The person which is having the book can give the book on rent for some days at a reasonable rent and can gain money and on the contrary, those who want to rent a book can communicate with the book giver through our portal. So, in general, our online book rental system will allow a user to rent a book online from the website. Once a book is rented it is no longer available to rent again by other users until the book is returned. The book which has already been rented will show the name of the renter and the date until when it is rented for the main objective of the project is to create an online book rental site that allows users to search and rent a book online based on title, author and subject. Using this Website, the user can rent a book online instead of going out to a book store and library spending a lot of their time meanwhile also ensuring giving them the opportunity to passively earn money in exchange for their books.

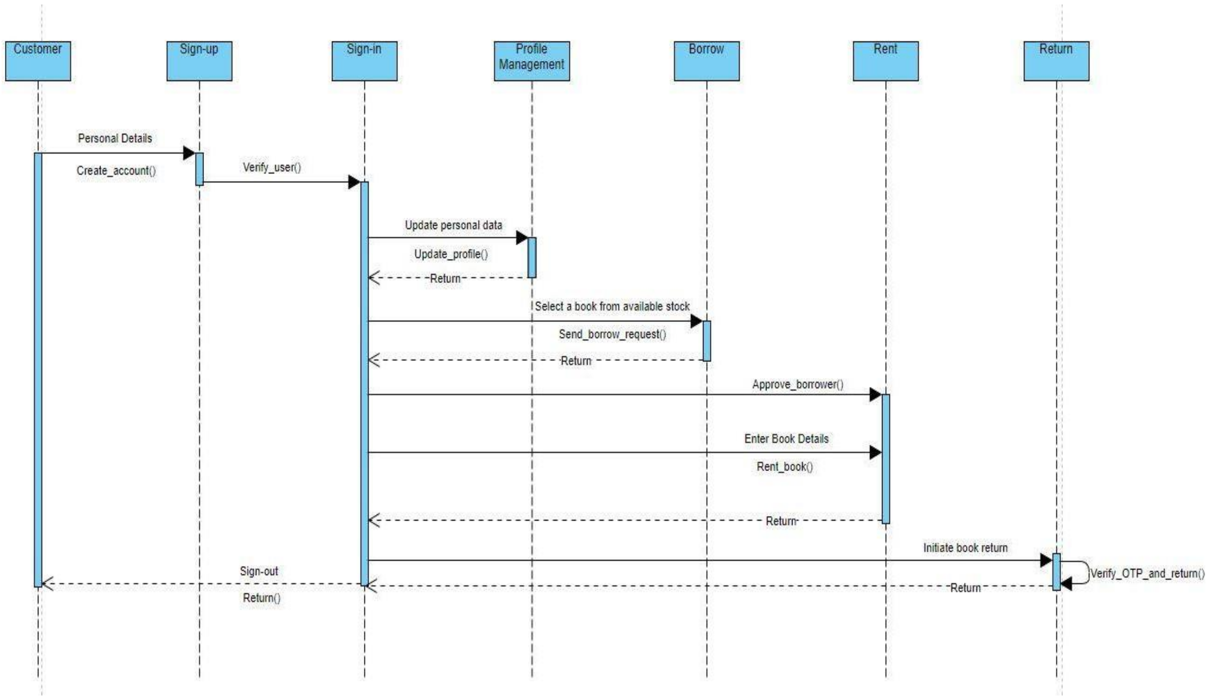
Problem Statement

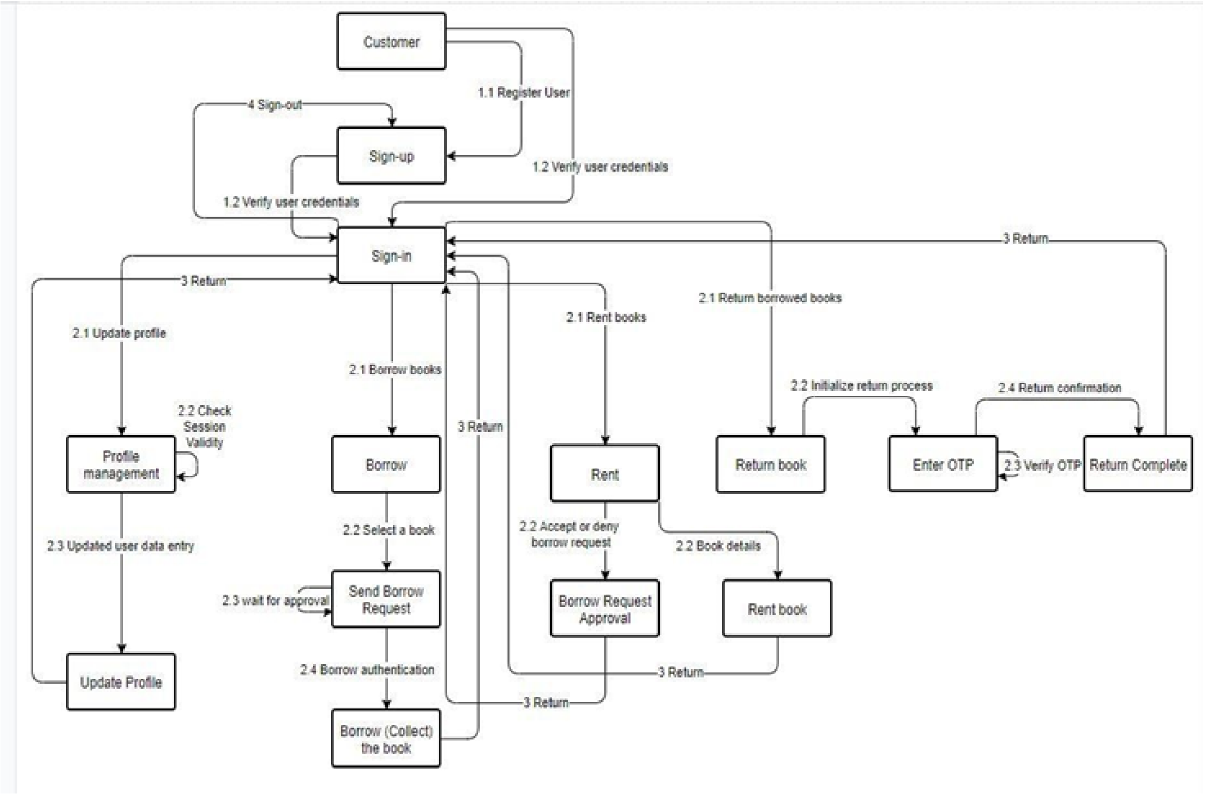
If a reader wants to read a book, he/she has to purchase the book, where they have to give much money or they can go to the library to lend a book where firstly they have to be a member and have to follow the library rules and regulations - like to return the book at the specified date that is given by the library. So, a lot of time and money gets wasted on these. With our portal situation becomes very easy. The person which is having the book can give the book on rent for some days at a reasonable rent and can gain money and on the contrary, those who want to rent a book can communicate with the book giver through our portal. So, in general, our online book rental system will allow a user to rent a book online from the website.

Reason

The main leverage of this project is the fact that many people like collecting books, including ourselves whereas many people don't have access to books for cheap, or do not particularly want to read the entire book but only a segment of that particular book. Also, many people like to read stuff in hardcopy format instead of softcopies. These two groups of people can massively benefit each other through the use of an application such as ours. Borrowing and renting such books makes a lot of sense to us as in most cases, after we read a particular book, we don't refer to most books again very often and they just accommodate with not a lot of use. Through our application, people can make their book available for rent for other people to borrow for some time which gives the renter a passive income source and the borrower a cheap alternative to read from a book instead of buying a new book every time. It also helps to create a community of people having similar interests.

Process Flow Diagram





Modules

Requirements and Frameworks: HTML, CSS, JavaScript, Node.js, Passport.js, Express.js, MongoDB (mongoose.js)

Database model creation

A database model shows the logical structure of a database, including the relationships and constraints that determine how data can be stored and accessed. Individual database models are designed based on the rules and concepts of whichever broader data model the designers adopt. Most data models can be represented by an accompanying database diagram.

Sign-in, Sign-up, Sign-out

Sign-in

Sign-in is for the session creation

Sign-up

Sign-up is for the registration of customers & lenders.

Sign-out

Sign-out is for ending an already existed session

User profile management

The User Profile Management feature provides central management for user-specific data and settings stored in the users' model.

Book rental system

Users can make the books available for rent from here for other people to borrow.

Book borrower system

Users can browse the books and send a borrow request.

Book return system

This is where the borrower will return the book and pay if any due is there.

Prototype Design

On the home page, the user has to login into their account. If they do not have an account then the user has to sign up to the portal. After signing up, the user is redirected to the sign-in page where the user credentials are verified. If the credentials are correct, the website creates a session cookie to keep the user signed in and prevents access to sign-in and sign-up pages to prevent malicious users from using their generated cookie to access other's accounts. Also, a middleware is configured to check if the user is trying to bypass login to access restricted pages and if they are; the user is redirected to the login page to log in with their credentials and session cookie. After signing in, the user can see his/her profile. Users can update their profiles here. From here user gets an option to perform one of many activities. Users can choose to make a book available for rent, borrow books, return the borrowed books, or either sign out.

If the user chooses to check the rental system, he will be redirected to another page that is composed of two components, the book rental system, and the history of previously rented books. In the book rental system, there will be a form in which details of the book which the user wants to give away for rent will be submitted and the book will be made available in the borrow section of the application. The history section of the page is mainly used for two tasks. Firstly, to show details of all transactions. Secondly, if any other user sends a request to borrow the book, the renter gets to choose if he wants to lend the book to that particular user or not. The book transaction gets through the initiation state only after the borrower is validated by the renter.

On the borrower section, the user can send a borrow request to the vendor of the book and browse through all the books which are up for borrowing. The restriction is that the user cannot borrow the book which he made available to borrow.

On the book return section, users can see the list of all the books he/she had ever borrowed from the portal along with the option to return the books he currently has not returned to its owner. For the transaction to be safely finalized, we will be implementing an OTP-like system. Whenever a book is made available for rent, a random number is also generated as an OTP which only the renter has access to. For the borrower to be able to complete the transaction, he has to enter that OTP for confirmation which the vendor will tell him/her. If the OTPs don’t match, the transaction won’t close. If the OTPs match, it can be deduced that both the borrower and vendor approve the completion of a transaction and therefore can be terminated.

The user can also manually sign out which expires the session cookie, otherwise, the session cookie will automatically expire after some time, making the user log in again for security reasons.

Database Details

Mongo DB used as database. MongoDB is a NoSQL database implying that the data stored in it is not structured and is flexible in nature, so any future model updates can easily be accommodated into the database.

We primarily used two NoSQL database models for our project which are:

1) User Model

2) Book Model

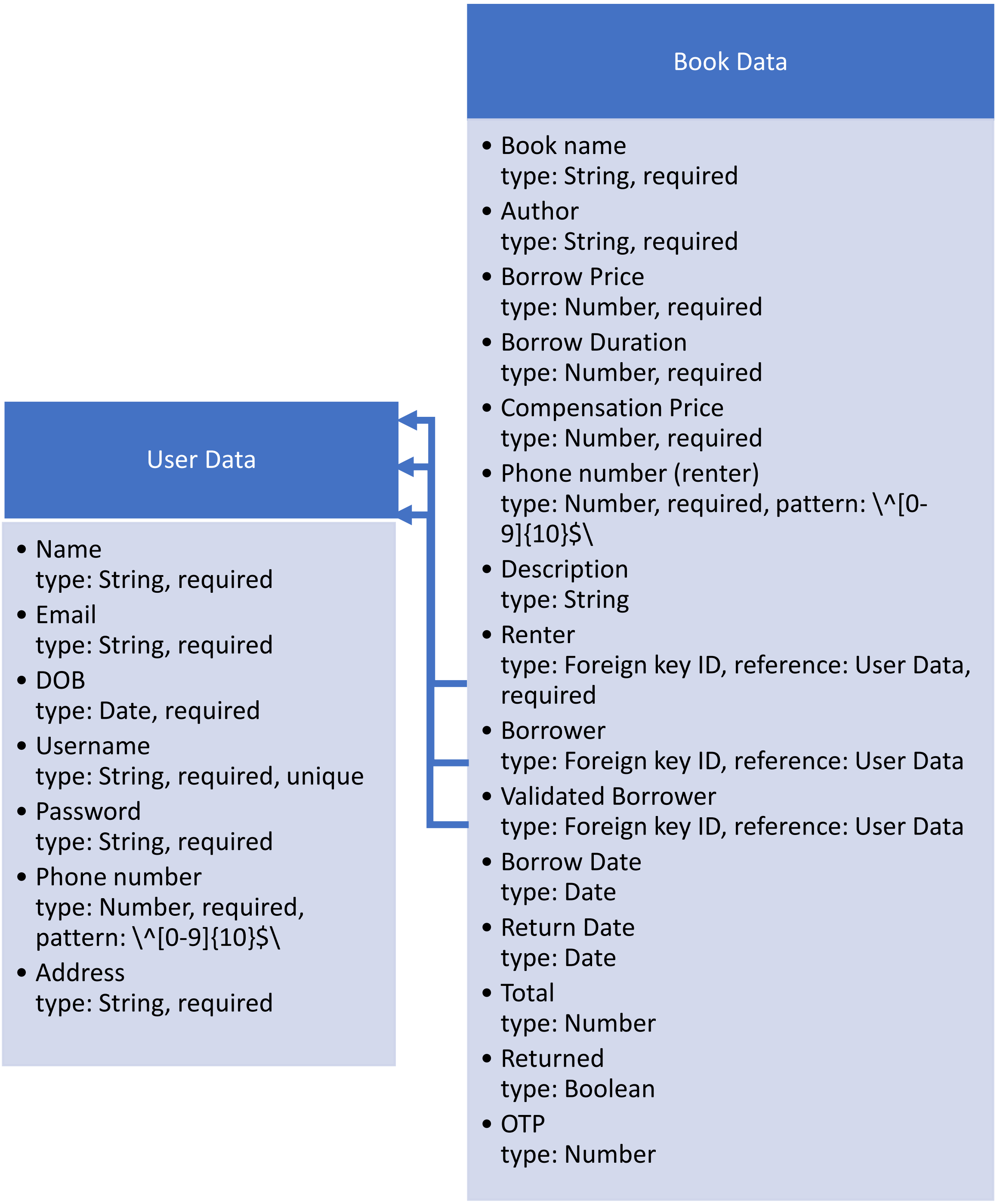
- User Model: This model contains all information about a particular user. Its model is as following:

```
Name: {
type: String,
required: true
},
E-mail: {
type: String,
required: true,
unique: true
},
DOB: {
type: Date,
required: true,
},
Username: {
type: String,
required: true,
unique: true
},
Password: {
type: String,
required: true
},
Phone no: {
type: Number,
required: true
},
Address: {
type: String,
required: true
}
```

- Books Model: This model contains about all incoming and outgoing books. All the current transactions and history is stored here. This model is used to link the users to transaction and maintain records. Its model is as following:

```
book name: {
type: String,
required: true
},
book author: {
```

```
type: String,
required: true
},
borrow price: {
type: Number,
required: true
},
borrow duration: {
type: Number,
required: true
},
compensation price: {
type: Number,
required: true
},
phone: {
type: Number,
required: true,
},
description: {
type: String
},
renter: {
type: Foreign Key to user model,
required: true
},
borrower: {
type: Foreign Key to user model
},
Validated borrower: {
type: Foreign Key to user model
},
borrow date: {
type: Date,
},
return date: {
type: Date,
},
total: {
type: Number
},
returned: {
type: Boolean,
default: false
},
OTP: {
type: Number,
default:0
}
```



User Data

- Name
  type: String, required
- Email
  type: String, required
- DOB
  type: Date, required
- Username
  type: String, required, unique
- Password
  type: String, required
- Phone number
  type: Number, required, pattern: \^[0-9]{10}$\
- Address
  type: String, required

Book Data

- Book name
  type: String, required
- Author
  type: String, required
- Borrow Price
  type: Number, required
- Borrow Duration
  type: Number, required
- Compensation Price
  type: Number, required
- Phone number (renter)
  type: Number, required, pattern: \^[0-9]{10}$\
- Description
  type: String
- Renter
  type: Foreign key ID, reference: User Data, required
- Borrower
  type: Foreign key ID, reference: User Data
- Validated Borrower
  type: Foreign key ID, reference: User Data
- Borrow Date
  type: Date
- Return Date
  type: Date
- Total
  type: Number
- Returned
  type: Boolean
- OTP
  type: Number

Conclusion

Book Look is an online web application where the customer can rent books online. Through a web browser, the customers can search for a book by its title or author, in general by this project one can rent the book from anywhere at any time. The convenience of an online book rental is a major consideration for most of us, especially those living in areas where access to a large bookstore is limited or is quite far away. Many books are too much expensive and for these types of books, this system is most preferable. If a book is too expensive, then one can rent that Book for some amount and can read that book by spending a very small amount of money. After reading the whole book they can return a book and another one can take Benefit of this type of expensive books. Also, Textbook rentals are great for when you only need the textbook for a small section of the course. In such cases, purchasing a book just for one chapter could be a waste of money. So overall by this system one can read the book of their choice at any time by spending a very small amount of money. Using our platform both parties problems will get solved. The person which is possessing any book will get some sort of rent price so he will get benefited and the person who wants to read the book can get a book at a much less price and should return that book as per the norms and conditions decided. The user can also give feedback on the condition of the book and convey its overall renting experience by giving ratings on a score of five and can add beneficial comments so that the buyers get to know the genuine sellers. This also prevents the buyers from fraud and hence creating the best experience.

References

The application is based on HTML, CSS and JavaScript with JavaScript specific frameworks such as:

Node.js (version), Passport.js, Express.js and mongoose.js (MongoDB)

cookie-parser: ^1.4.5,

ejs: ^3.1.6,

express: ^4.17.1,

express-session: ^1.17.2,

mongoose: ^5.13.7,

nodemon: ^2.0.12,

passport: ^0.4.1,

passport-local: ^1.0.0,

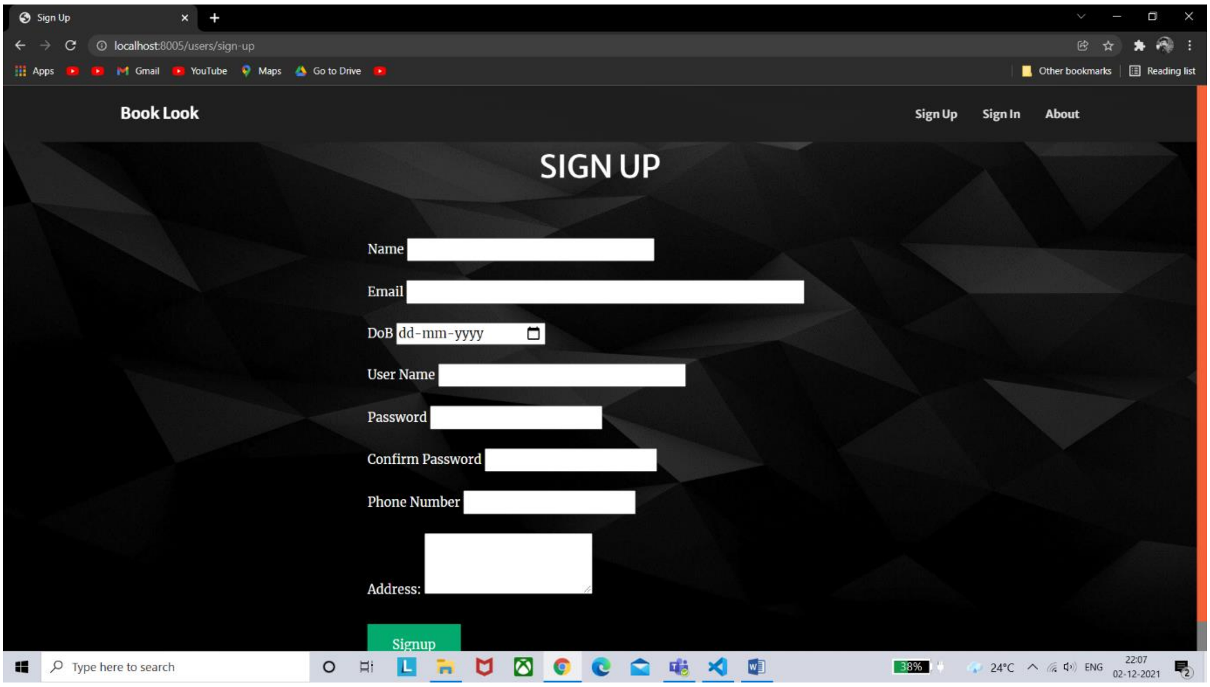
node: ^6.14.13

1. Online Vehicle Rental System (https://ijsret.com/wp-content/uploads/2020/05/IJSRET_V6_issue3_362.pdf)

2. Implementing a Computerized Film Rental System (https://www.tandfonline.com/doi/abs/10.1300/J107v05n02_05?journalCode=wjcl20)
3. Practical development of information system in business context: book rental system for N. Book Company (https://repository.au.edu/server/api/core/bitstreams/3c956109-32ec-4310-b71c-549bf43fb4fa/content)

Snapshots

SIGN-UP



Sign-up page consists of a form which is filled by the customer to get registered on the website. The form consists of 8 fields:

- Name (string)

  The name can contain any character

- Email (string)

  The email should be in a proper email address format

- Date of birth (date)

  The customer has to add their date of birth using the calendar.

- User Name (string)

  The username can contain any character but it has to be unique.

- Password (string)

  The password can contain any character

- Confirm Password (string)

  The confirm password input should match the password.

- Phone Number (string)

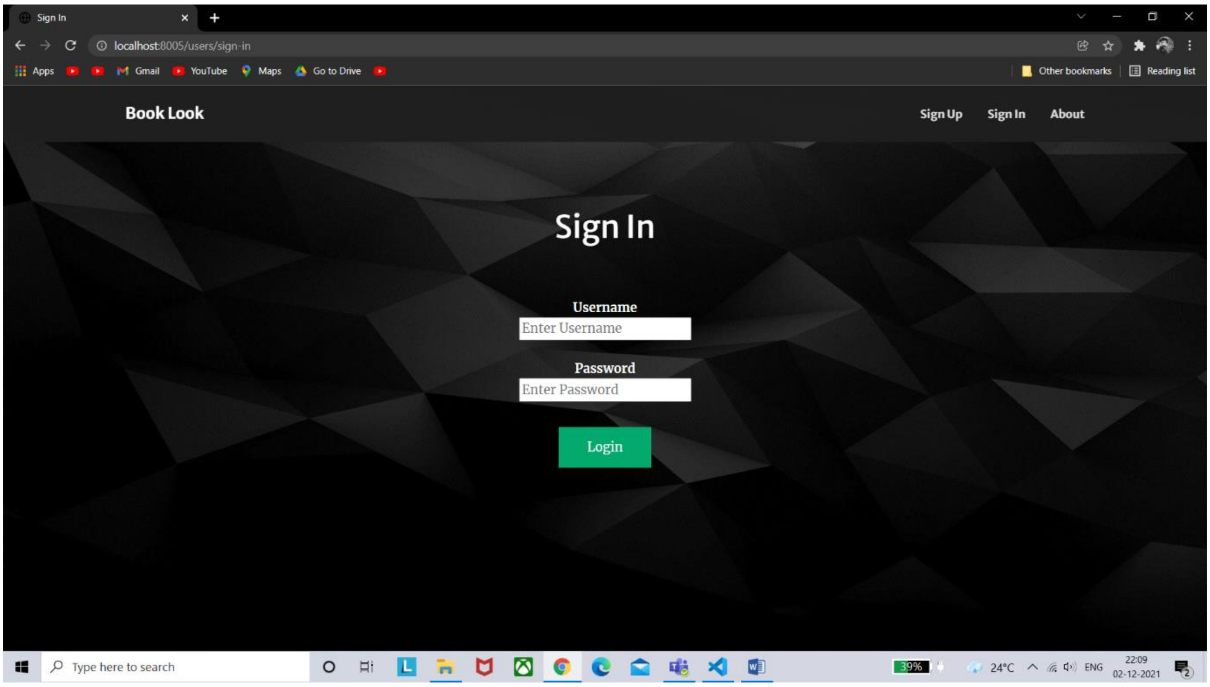
  (+91) by default, enter a ten-digit phone number (no landline).

- Address (string)

  The address should contain the home address of the customer.

The customer has to fill each field of the form and then click on the signup button, after clicking on the signup button the customer will be registered as a user and their information will be stored in the database, and then the customer will be redirected to sign-in page.

SIGN-IN



Sign-in page consists of a form which is filled by the user to access their account. The form consists of two fields:

- Username (string)

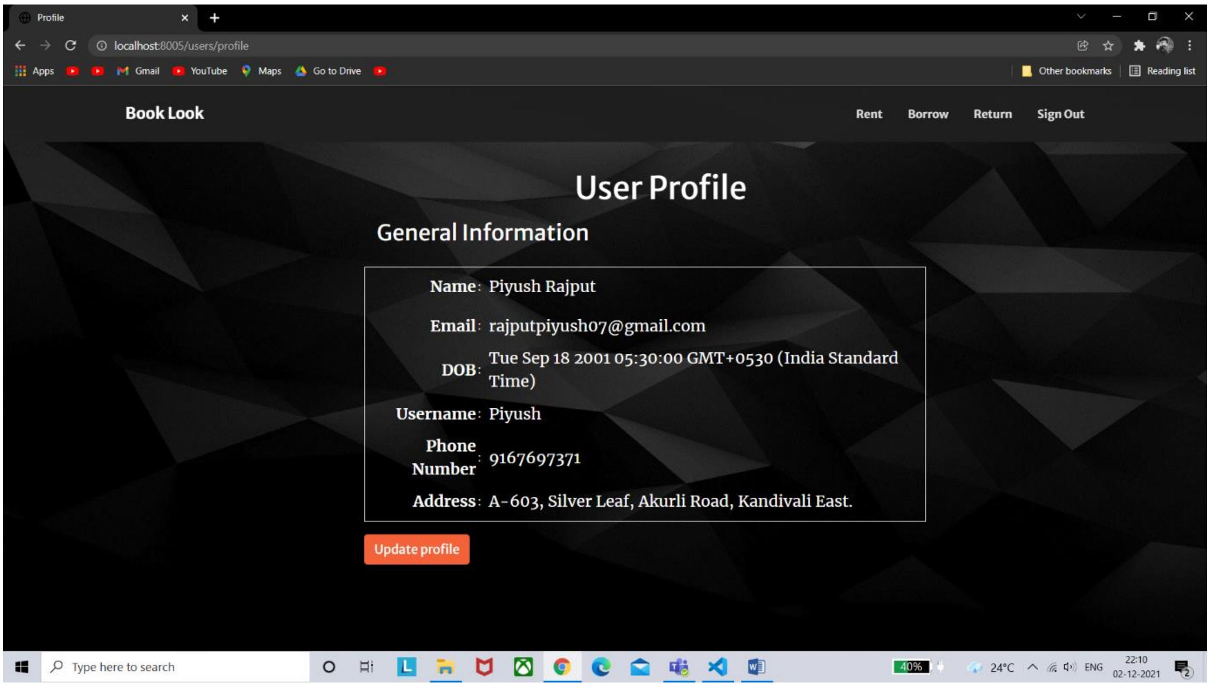
  The username can contain any character but it has to be unique.

- Password (string)

  The password can contain any character

The user has to fill the form and then click on the signin button, after clicking the signing button the user credentials will be verified, if the user is registered in the system, a session will be created and they will be redirected to their profile page otherwise if the credentials are not in the system then they will be redirected to the signin page.

PROFILE



User Profile

General Information

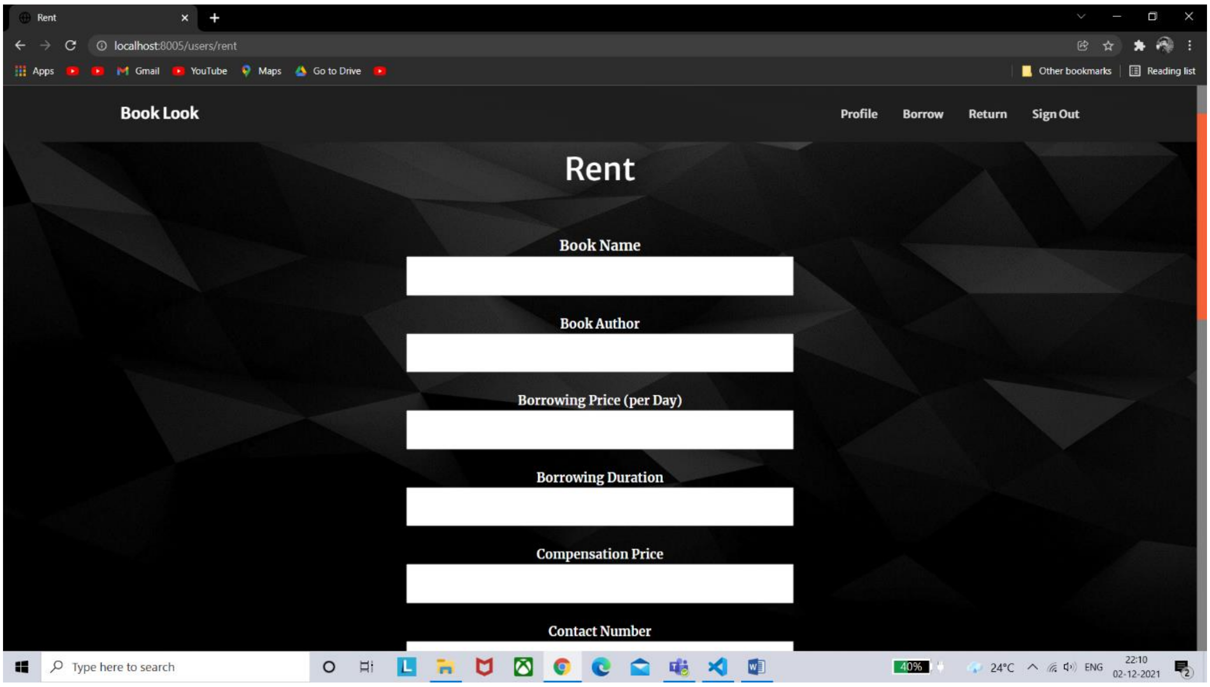
Name:	Piyush Rajput
Email:	rajputpiyush07@gmail.com
DOB:	Tue Sep 18 2001 05:30:00 GMT+0530 (India Standard Time)
Username:	Piyush
Phone Number:	9167697371
Address:	A-603, Silver Leaf, Akurli Road, Kandivali East.

The profile page consists of a table and an update profile button. The table contains the information of the user such as name, email address, date of birth username, phone number and address.

If the user clicks on the update profile button, they will be redirected to the update profile page where the user can update their information (given in the table).

On the navigation bar there are three buttons rent, borrow, return and sign-out. If the user clicks on the rent button, then he/she will get redirected to the rent page, similarly for return and borrow the user will get redirected to the respective pages. If the user clicks on the signout button, the session will be deleted and they will get redirected to the signin page.

RENT



Rent

Book Name

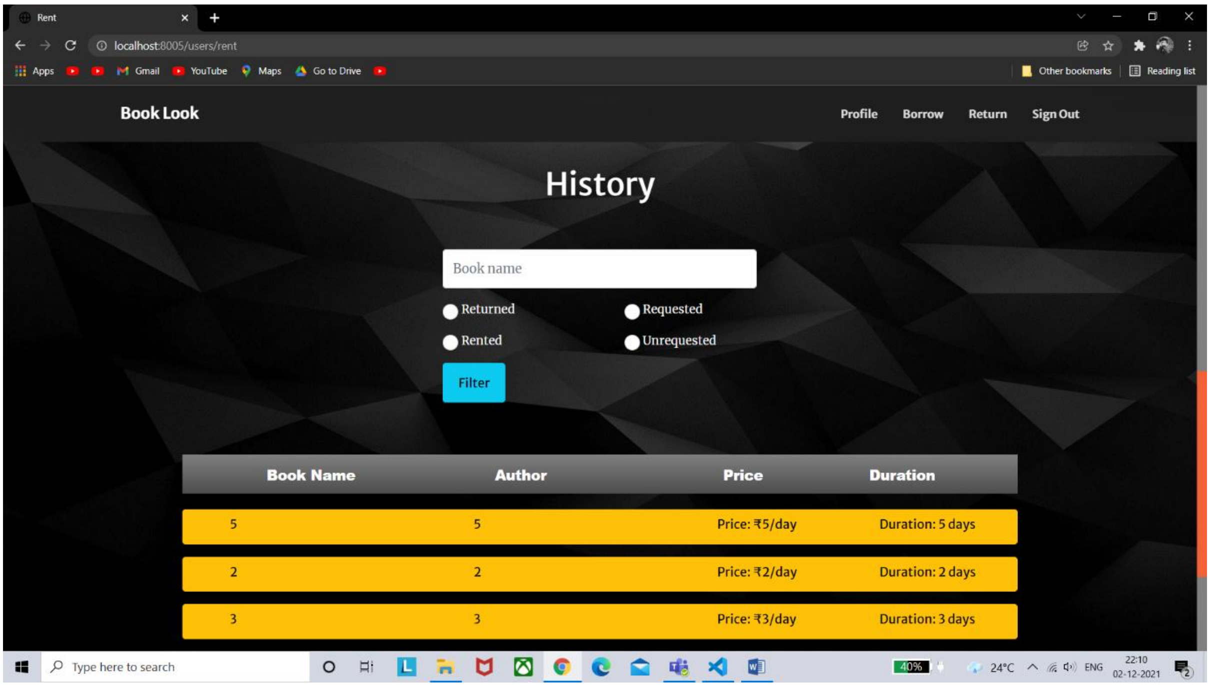
Book Author

Borrowing Price (per Day)

Borrowing Duration

Compensation Price

Contact Number



History

Book name

○ Returned ○ Requested

○ Rented ○ Unrequested

Book Name	Author	Price	Duration
5	5	Price: ₹5/day	Duration: 5 days
2	2	Price: ₹2/day	Duration: 2 days
3	3	Price: ₹3/day	Duration: 3 days

The rent page consists of a form and a rent history, the form is used to enter the details of the book the user wants to rent and the rent history is used to see the books rented by the user in the past.

The form contains seven fields:

- Book Name (string)

  The book name can contain any only alphabets and numbers.

- Book Author (string)

  The book author can contain any character.

- Borrowing Price per day (string)

  The borrowing price should only contain numbers.

- Borrowing Duration (string)

  The borrowing duration contains the amount of time the book will be rented for. It should only contain alphabets and numbers.

- Compensation Price (string)

  The compensation price should only contain numbers.

- Contact Number (string)

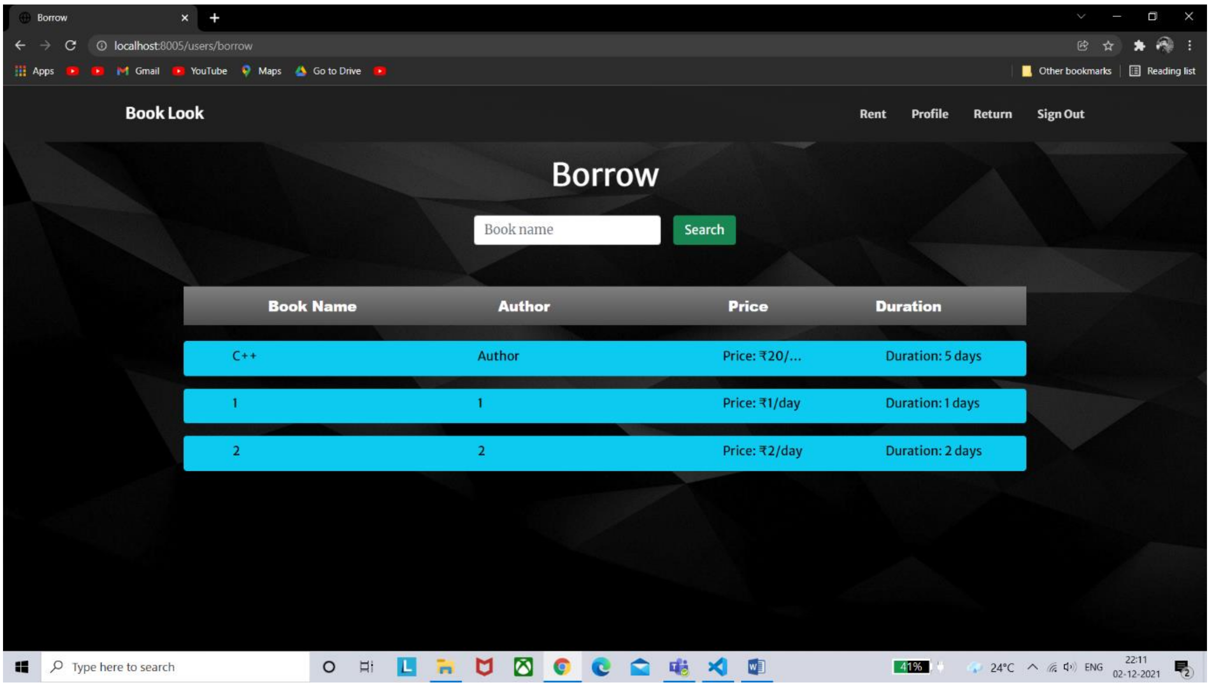
  (+91) by default, enter a ten-digit number.

- Description (string)

  The description can contain any character.

The rent history contains the history of the books rented by the user. The user can validate the borrower and it also contains an OTP field. The OTP is used by the borrower to return the book which is randomly generated and provided by the owner of the book (renter).

BORROW

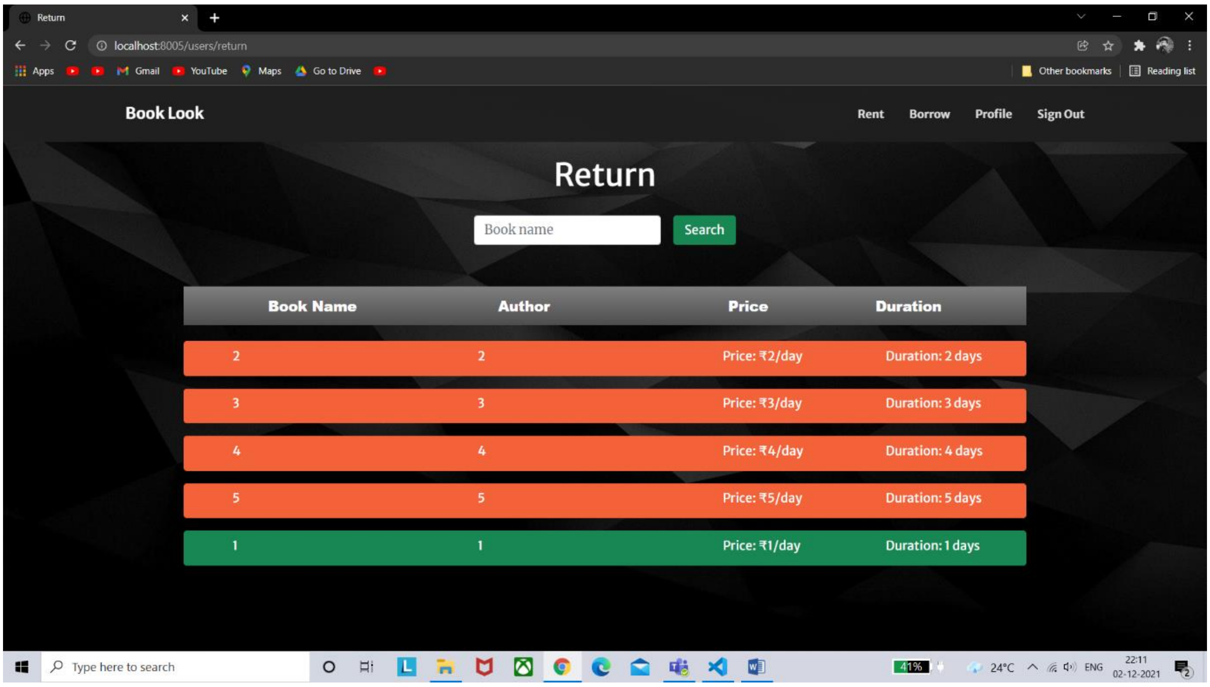


Borrow

Book Name	Author	Price	Duration
C++	Author	Price: ₹20/...	Duration: 5 days
1	1	Price: ₹1/day	Duration: 1 days
2	2	Price: ₹2/day	Duration: 2 days

The borrow page consists the borrow information of the books. It also contains a borrow button which sends the request to the renter for borrowing the book. It shows all the books which are available for borrowing, it shows all the details of the books such as book name, author name, price, duration, name and contact number of the renter and a compensation price if the book is damaged.

RETURN



Return

Book Name	Author	Price	Duration
2	2	Price: ₹2/day	Duration: 2 days
3	3	Price: ₹3/day	Duration: 3 days
4	4	Price: ₹4/day	Duration: 4 days
5	5	Price: ₹5/day	Duration: 5 days
1	1	Price: ₹1/day	Duration: 1 days

The return page contains the information of the books borrowed by the user; it also contains a section return which allows the user to return the book to its owner. To return the book the user has to enter the OTP given by the owner of the book. The book can only be returned after submitting the correct OTP by clicking on the submit button.

Source Code

Link: https://github.com/Gaurav1020/BookLook