University of Asia Pacific

CSE 302: Object Oriented Programming II Lab: Visual and Web Design

Project Documentation

BookLoop Library

Submitted to

Durjoy Mistry
Lecturer, Department of CSE, University of Asia Pacific

Submitted by

Group Name: Status 200 OK Group Number: 01

Name	Registration ID
Nafis Fuyad Niloy	22201223
Md. Jubaer	23101008
Md. Osman Goni	23101012
Aeysha Tabassum	23101014
Arannamoy Mondal	23101024

Introduction:

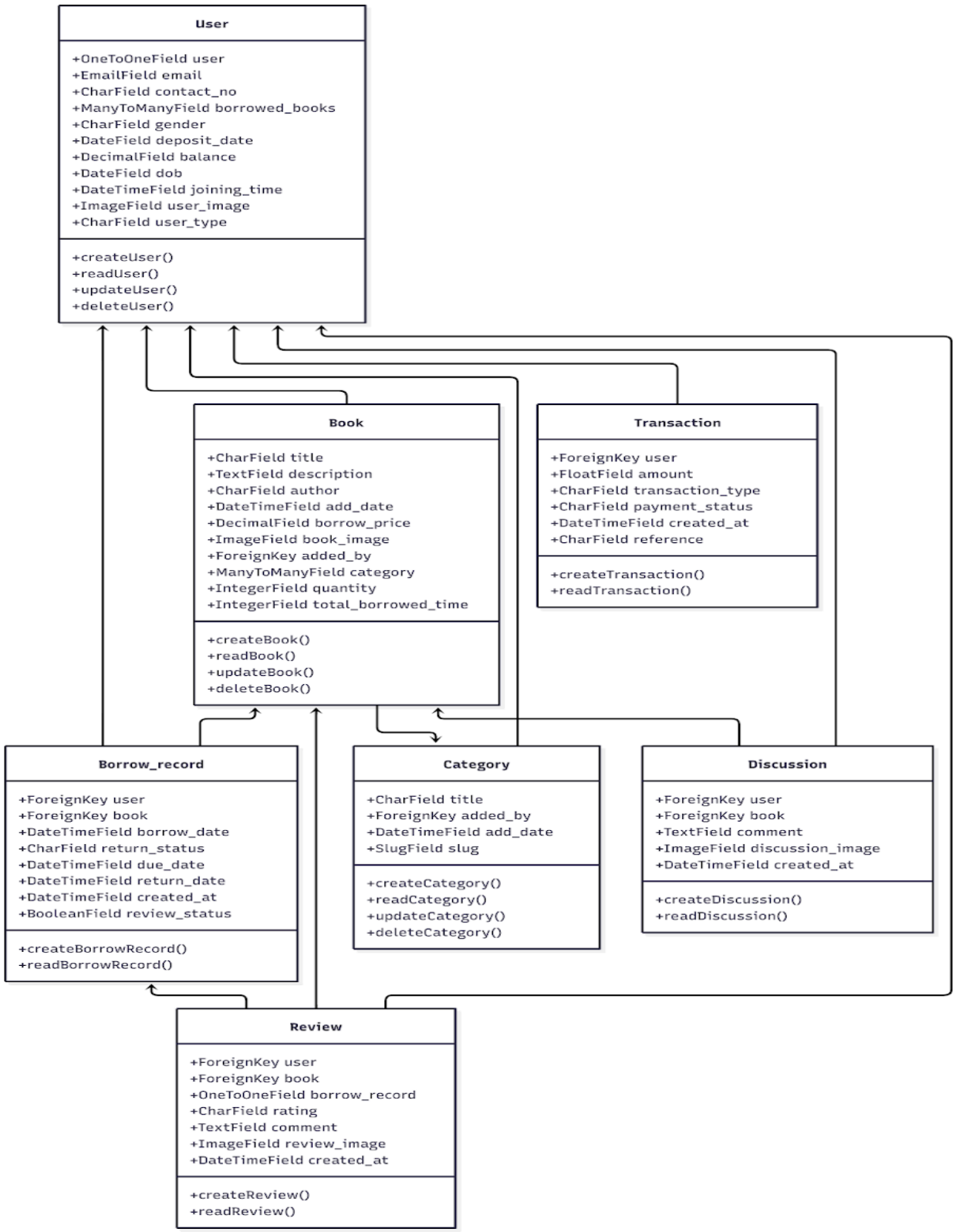
Description:

BookLoop is a responsive web-based library management system built with Django, Jinja Templates, Tailwind CSS, and DaisyUI. It supports three user roles—Reader, Administrator, and Superuser. Readers can borrow books, review them, and view their transaction and borrowing history. Administrators can additionally manage books and categories, while the Superuser can promote readers to administrators.

Motivation:

The project aims to create a hassle-free, cashless, and open-source library system that simplifies book borrowing, enhances transparency for both readers and authorities, and promotes knowledge sharing within a connected digital community.

Class Diagram



1. **Book:** Represents a book in the library.
    - **FKs:** added_by → who added the book; category → related book categories.
2. **Borrow_record: Tracks each book borrowing activity.**
    - **FKs**: user → borrower; book → borrowed book.
3. **Category: Defines book genres or groups.**
    - **FK:** added_by → who created the category.
4. **Discussion: Stores user comments on books.**
    - **FKs:** user → commenter; book → discussed book.
5. **Review: Holds user reviews and ratings.**
    - **FKs:** user → reviewer; book → reviewed book; borrow_record → related borrow instance.
6. **Transaction: Logs user payments and deposits.**
    - **FK:** user → who made the transaction.
7. **User: Extends Django's base user with library info.**
    - **FKs:** user → base Django user; borrowed_books → books currently borrowed.

Project Structure:

```
├── BookLoop
│   ├── asgi.py
│   ├── __init__.py
│   ├── settings.py
│   ├── urls.py
│   ├── views.py
│   └── wsgi.py
├── books
│   ├── admin.py
│   ├── apps.py
│   ├── __init__.py
│   ├── models.py
│   ├── templates
│   │   ├── add_book_form.html
│   │   ├── all_books.html
│   │   ├── book.html
│   │   └── books.html
│   ├── test.py
│   ├── tests.py
│   ├── urls.py
│   └── views.py
├── borrow_records
│   ├── admin.py
│   ├── apps.py
│   ├── __init__.py
│   ├── models.py
│   ├── templates
│   │   └── borrow_record.html
│   ├── tests.py
│   ├── urls.py
│   └── views.py
├── categories
│   ├── admin.py
│   ├── apps.py
│   ├── __init__.py
│   ├── models.py
│   ├── templates
│   │   └── category.html
│   ├── tests.py
│   ├── urls.py
│   └── views.py
├── CONSTANT.py
├── discussions
│   ├── admin.py
│   ├── apps.py
│   ├── __init__.py
│   ├── models.py
│   ├── tests.py
│   ├── urls.py
│   └── views.py
├── .env.sample
├── .gitignore
├── global_templates
│   ├── base.html
│   ├── carousel.html
│   ├── failed_modal.html
│   ├── footer.html
│   ├── index.html
│   ├── navbar.html
│   ├── review_carousel.html
│   ├── temporary.html
│   └── transaction_modal.html
├── manage.py
├── media
│   ├── book_image
│   ├── discussion_image
│   ├── review_image
│   └── user_image
├── Readme.md
├── requirements.txt
├── reviews
│   ├── admin.py
│   ├── apps.py
│   ├── __init__.py
│   ├── models.py
│   ├── templates
│   │   └── review_history.html
│   ├── tests.py
│   ├── urls.py
│   └── views.py
├── static
│   └── style.css
├── transactions
│   ├── admin.py
│   ├── apps.py
│   ├── __init__.py
│   ├── models.py
│   ├── templates
│   │   └── transactions_history.html
│   ├── tests.py
│   ├── urls.py
│   └── views.py
└── users
    ├── admin.py
    ├── apps.py
    ├── forms.py
    ├── __init__.py
    ├── models.py
    ├── templates
    │   ├── deposit_form.html
    │   ├── login_form_fun.html
    │   ├── login_form.html
    │   ├── payment_cancel.html
    │   ├── payment_fail.html
    │   ├── profile.html
    │   ├── signup_form.html
    │   ├── update_profile.html
    │   └── update_profile_t.html
    ├── tests.py
    ├── urls.py
    └── views.py
```

Installation and Setup:

1. **Clone the Repository:** First, clone the project repository from GitHub:

   ```
   git clone <repository_url>  &&  cd BookLoop
   ```

2. **Set up Python Environment:** Ensure that Python (version 3.8 or higher) is installed. Then, create and activate a virtual environment:

   ```
   python3 -m venv .venv
   source .venv/bin/activate       # For Linux/Mac
   .venv\Scripts\Activate.ps1      # For Windows
   ```

3. **Configure Payment Gateway:** Create a Sandbox Account on SSLCommerz obtain your sandbox API credentials. Create an environment file, copy the sample environment file, and rename it to .env:

   ```
   cp .env_sample .env
   ```

   Then, fill in the necessary variables such as SECRET_KEY, Store_ID, Store_Password, Issandbox, SSLZ_URL, and SSLCommerz sandbox credentials.

4. **Apply Database Migrations and Create Superuser:**

   ```
   python manage.py migrate && python manage.py createsuperuser
   ```

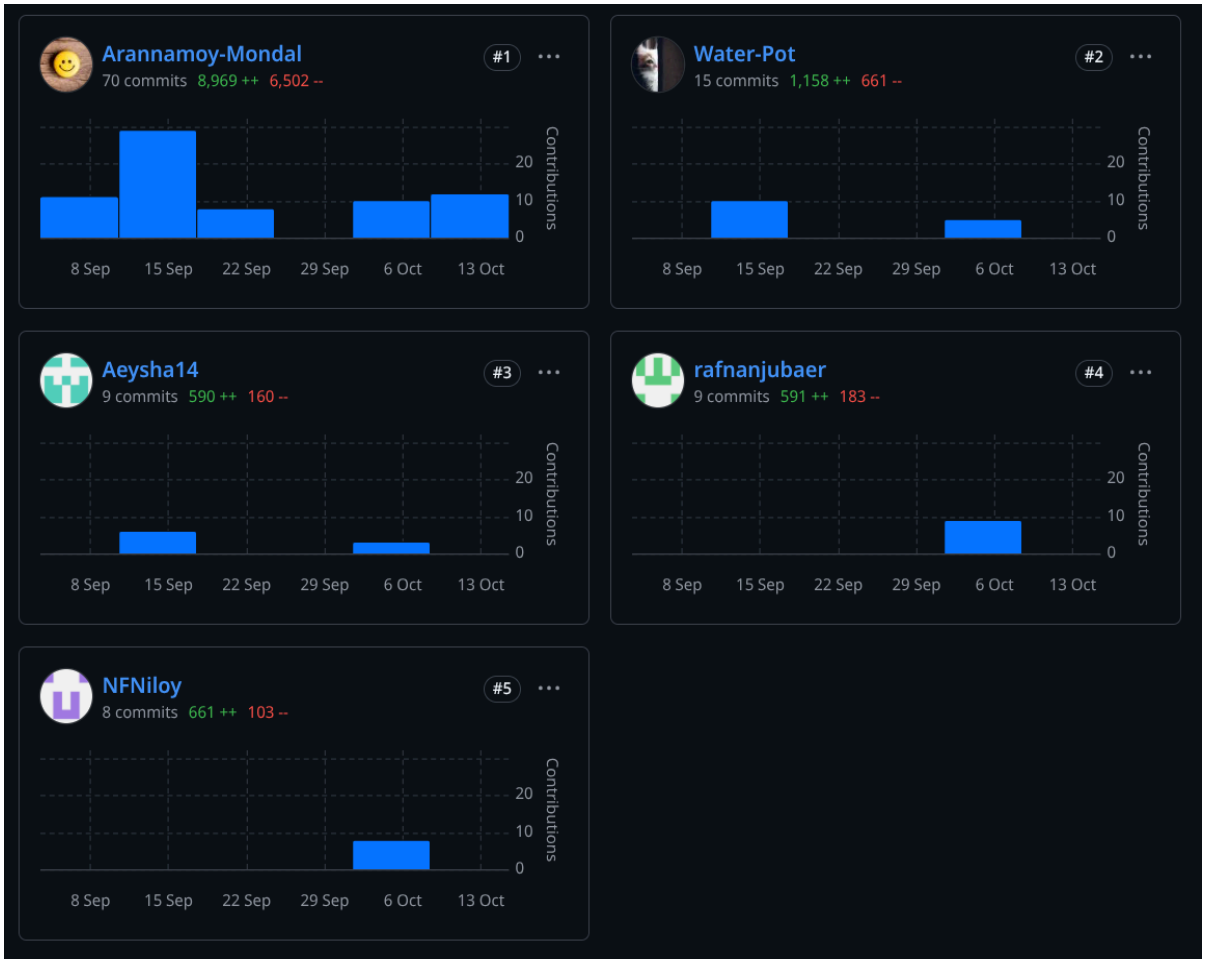
5. **Run the Development Server:**

   ```
   python manage.py runserver 0.0.0.0:8000
   ```

   The project will now be accessible at http://localhost:8000

Contribution

- Public GitHub link—
  https://github.com/Arannamoy-Mondal/BookLoop
- Screenshot of the contribution of each member:



Contribution of each member

SL	Full Name	Reg ID	Github username	Number of Commits	No of Lines added	No of Lines deleted
1.	Nafis Fuyad Niloy	22201223	NFNiloy	8	661	103
2.	Md. Jubaer	23101008	rafnanjubaer	9	591	183
3.	Md. Osman Goni	23101012	Water-Pot	15	1158	661
4.	Aeysha Tabassum	23101014	Aeysha14	9	590	160
5.	Arannamoy Mondal	23101024	Arannamoy-Mondal	66	8969	6502

Project Summary Table

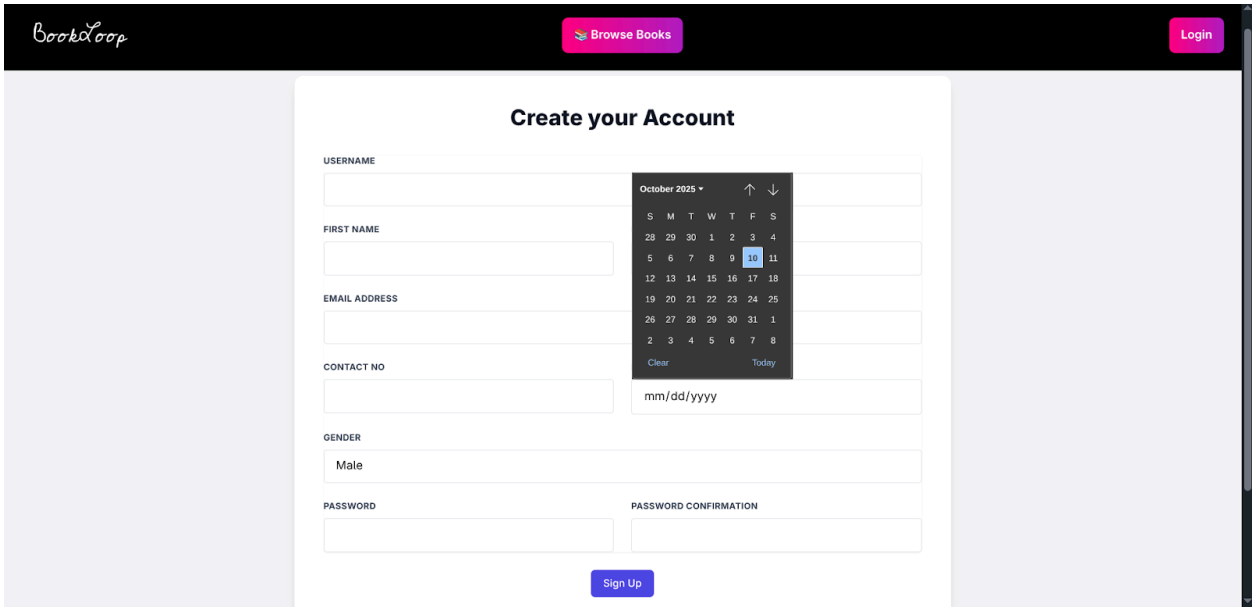
Components	Required	This project	Summery
Number of Apps	3	7	Name of the apps: users, books, categories, borrow_records, transactions, discussions, reviews
Number of Classes	7	7	Name of the classes: User, Book, Category, Borrow_record, Transaction, Discussion, Review
Number of foreign keys	2	11	Foreign keys are used in the following 6 classes, and they are - Book, Category, Borrow_record, Transaction, Discussion, Review
Use of Media	2	4	Media is used in the following 4 classes, and they are - User, Book, Discussion, Review

CEP Justification Table

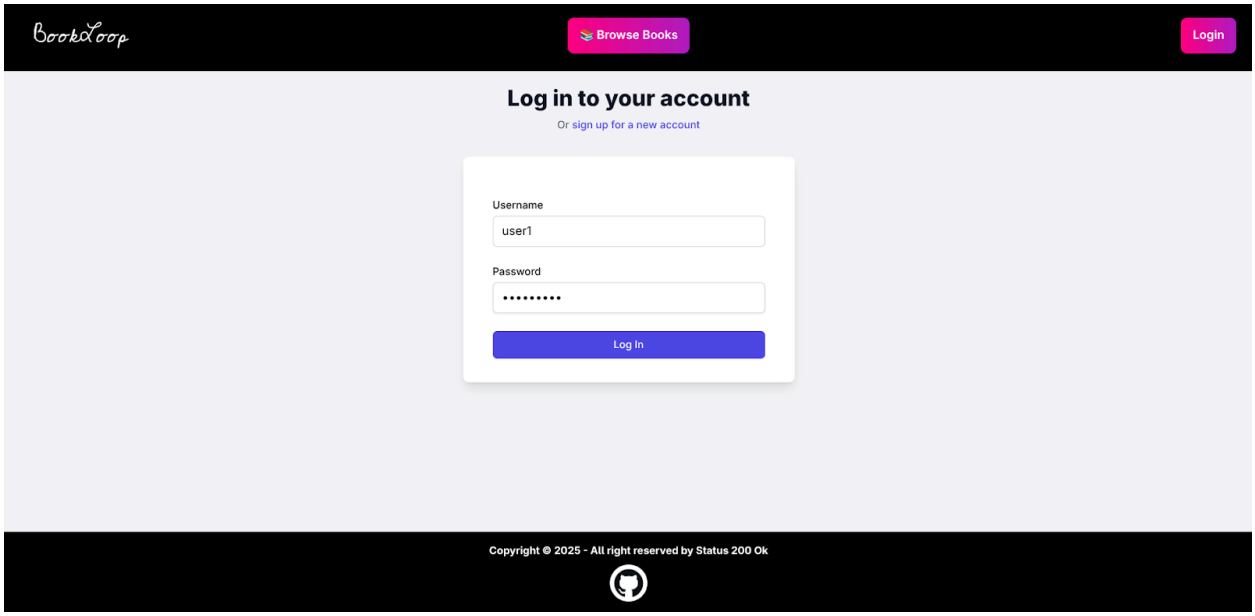
Aspect	Attribute	Justification for Django E-commerce Project	CLO	PLO
Knowledge (K)	**K2** – Mathematics, numerical analysis, information science	Calculating book borrowing prices, fine calculation and handling user transactions accurately.	CLO1	PLO1
	K3 – Engineering fundamentals	Understanding web architecture and data relationships between users, books, and transactions.	CLO1	PLO1
	K4 – Specialist knowledge	Application of Django ORM, secure SSL payment gateway integration.	CLO1, CLO3	PLO1, PLO5
	K6 – Knowledge of practice	Version control and project management were performed using **Git** for maintaining code integrity, tracking changes, and enabling efficient collaboration throughout the development process.	CLO2, CLO3	PLO3, PLO5
	K7 – Role of engineering in society	Promotes digital literacy, fair access to educational resources, and ensures ethical data handling for users.	CLO4	PLO6
Problem (P)	**P1 (mandatory):** Requires in-depth engineering knowledge	Involves integration of user authentication, transactions, category mapping, and database relations using multiple knowledge areas (K2–K7).	CLO1	PLO1
	P2: Conflicting requirements	Trade-offs between fast data access and secure payment/storage; optimizing book search speed vs. server cost.	CLO2	PLO2
	P3: No obvious solution	Designing dynamic borrow/return flow, handling overdue penalties, and preventing double borrowing requires creative logic.	CLO2, CLO7	PLO2, PLO3, PLO10
	P6: Multiple stakeholders	Students, librarians, administrators, and payment gateway providers interact through different modules.	CLO4, CLO6	PLO6, PLO9

	P7: Interdependent parts	Book, Borrow_record, Review, Discussion, and Transaction systems are tightly linked and depend on one another.	CLO2	PLO3
Activities (A)	**A1:** Diverse resources	Requires cloud hosting, database servers, the Django framework, SSL payment API, and admin/user collaboration.	CLO2, CLO6	PLO3, PLO9
	A2: Conflicting issues	Balancing data security, payment accuracy, and quick response for borrowing and returning operations.	CLO2, CLO4	PLO2, PLO6
	A3: Creative use of knowledge	Applying analytics to track borrowing trends, predict popular books, and enhance user experience with recommendations.	CLO2, CLO7	PLO3, PLO10

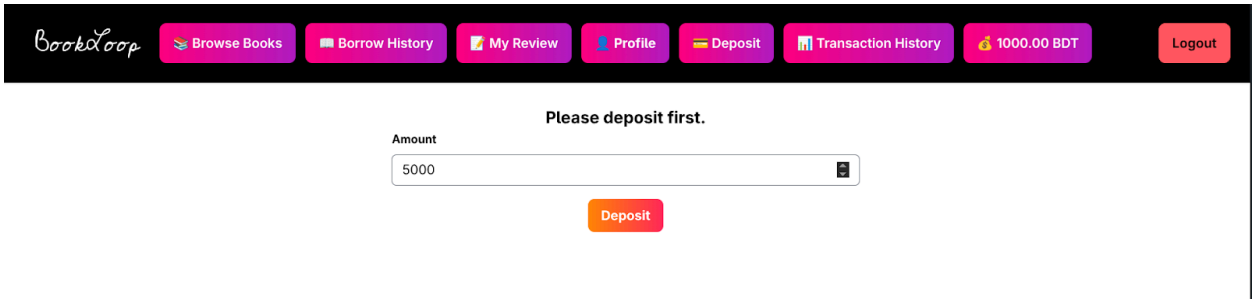
Screenshot



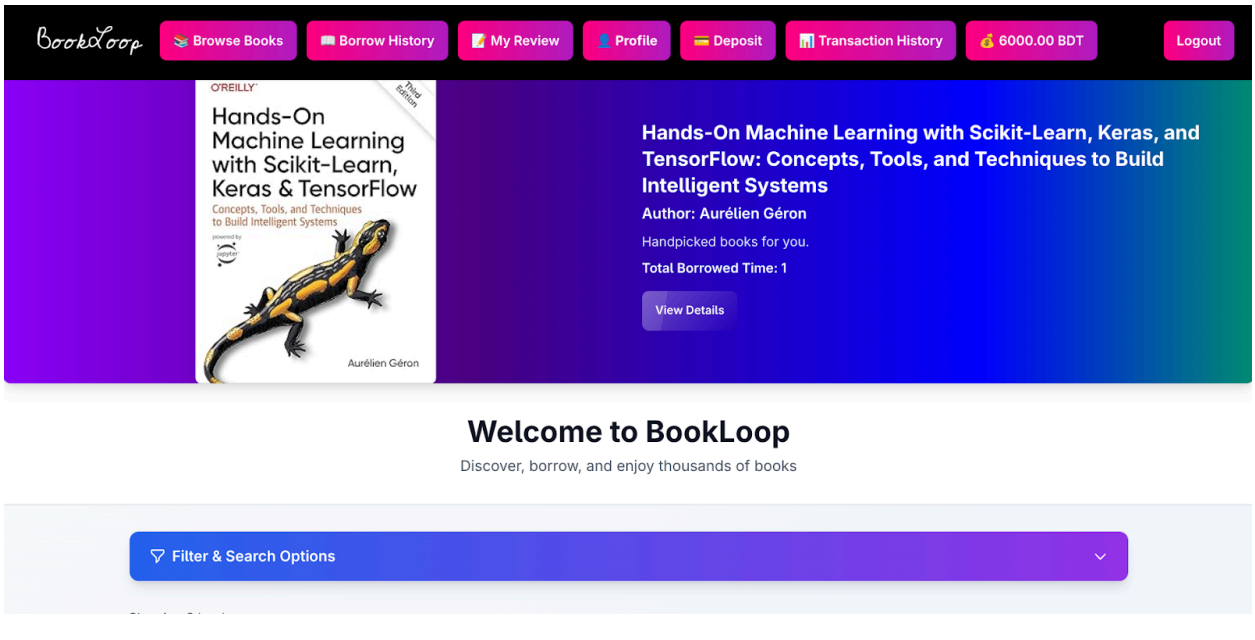
Sign Up



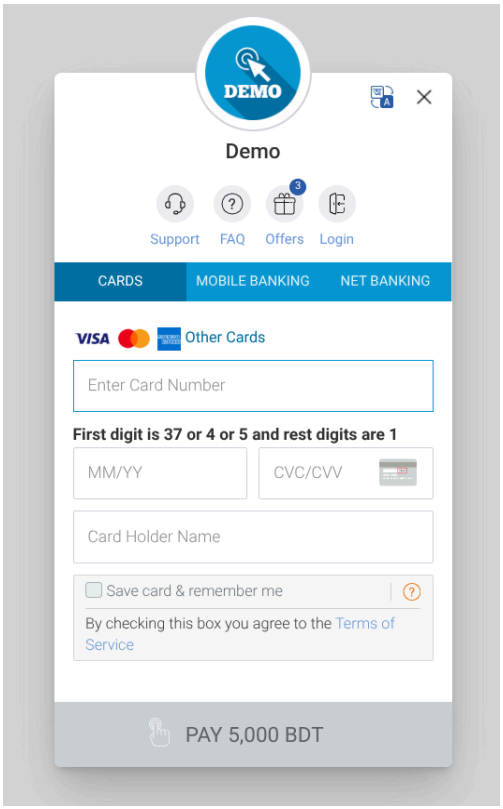
Login



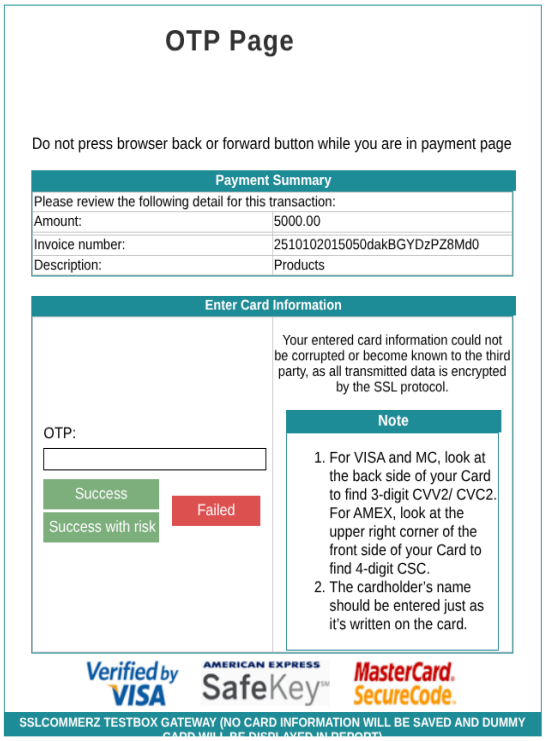
Deposit



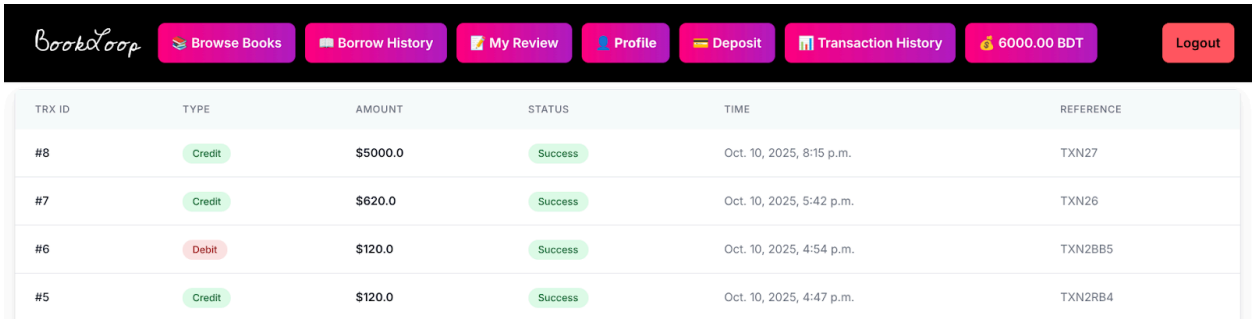
Homepage



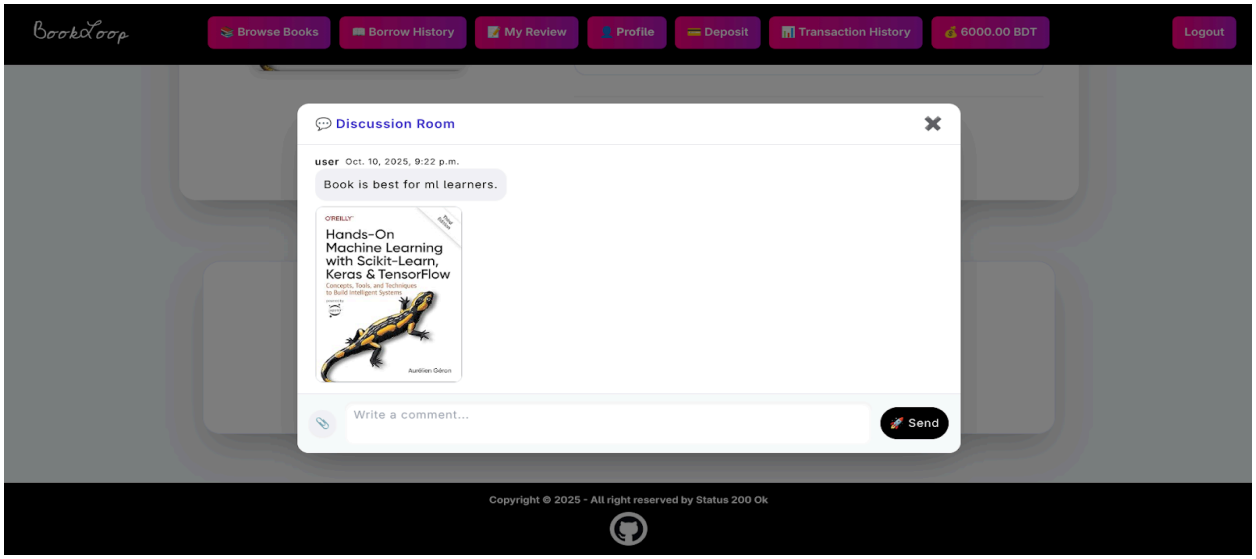
Payment Gateway



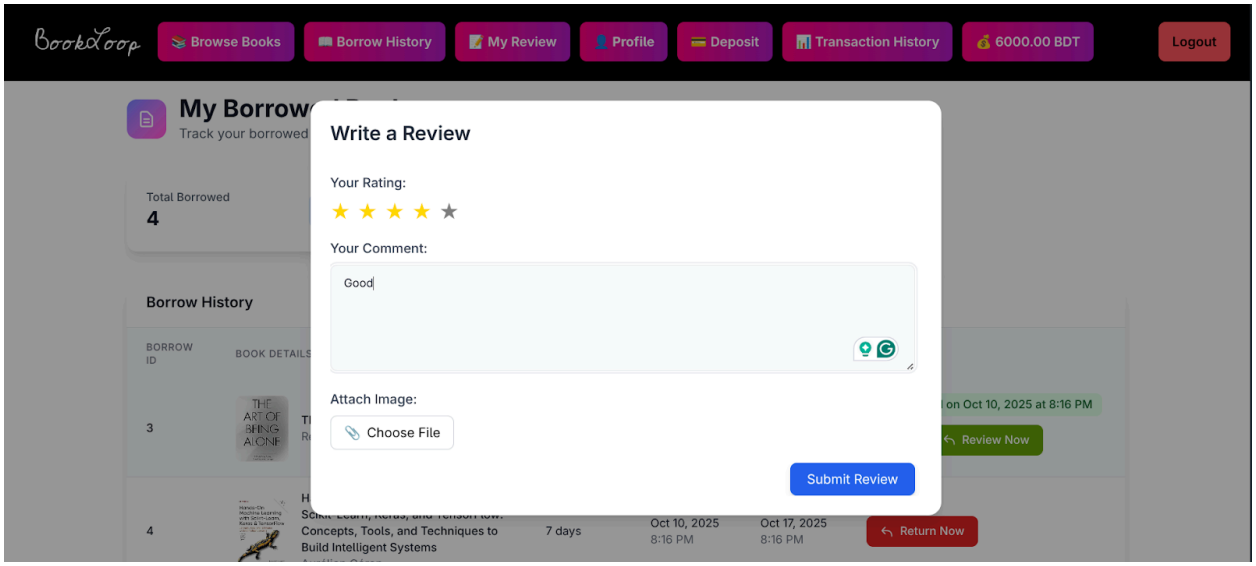
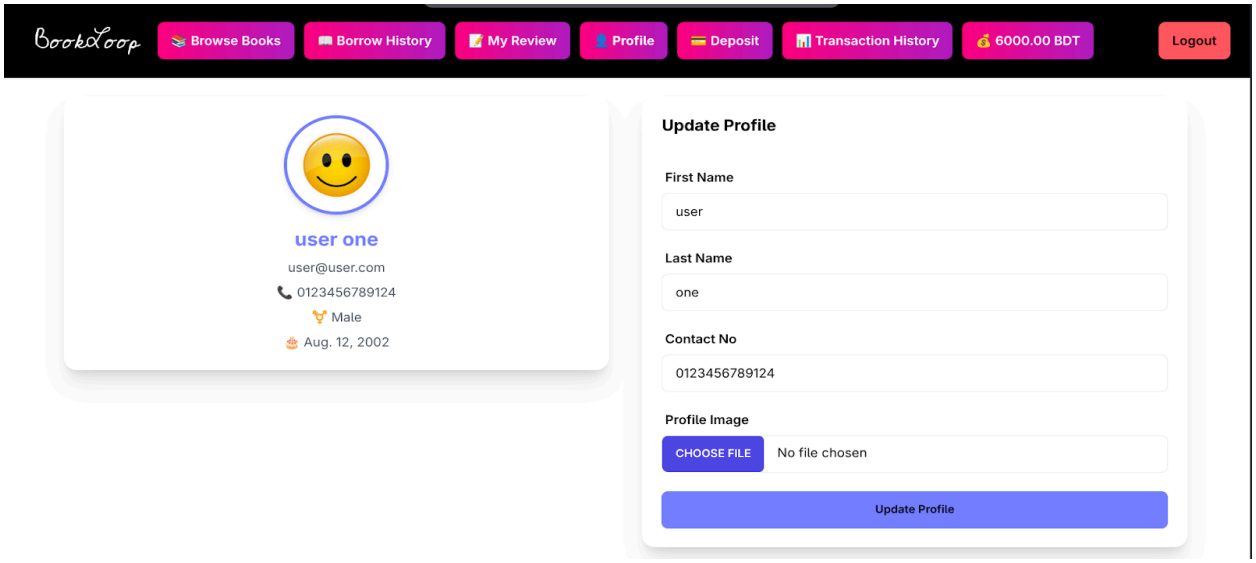
Otp verification



Transaction History



Discussion Room On A Specific Book



Write Review