ScholarLendingPortal API Documentation

Team Members (Group 24) and Contributions :

Team Member	Contribution
Yash Jha	Backend setup, Django API integration, documentation consolidation
Raju Kartik Jha	Frontend React components and AI prompt generation
Gokul G Gowda	Testing, debugging, and database schema design
Akshat Mishra	UI design and user flow (Student, Staff, Admin dashboards)

Project Overview:

ScholarLendingPortal is a full-stack web-based equipment lending system designed for schools. It enables role-based access for students, staff, and administrators to manage borrowing, approval, and return of shared equipment such as lab instruments, sports kits, and cameras.

Architecture Diagram:

React Frontend → Django REST API → SQLite Database

Database Schema:

Table	Key Fields	Relationships
User	id, username, role	1:N with BorrowRequest
Equipment	id, name, category, available	1:N with BorrowRequest & MaintenanceLog

Table	Key Fields	Relationships
BorrowRequest	id, equipment_id, user_id, status	FK to User & Equipment
MaintenanceLog	id, equipment_id, issue, resolved	FK to Equipment

Setup Documentation

Frontend:

Navigate to the frontend/ScholarLendingPortal folder, install npm and start the npm using `npm start`

If it doesn't start, then follow below steps,
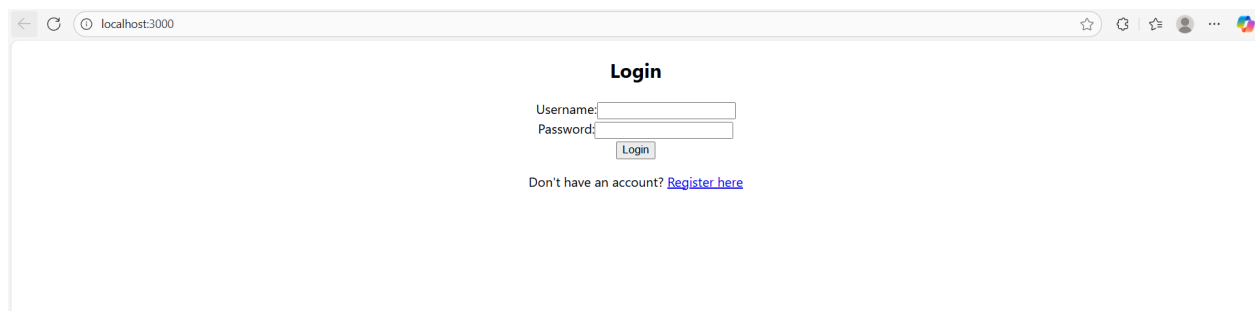
```
rm -rf node_modules
```

```
rm package-lock.json
```

```
npm install
```

```
npm start
```

Once npm is started, frontend application will automatically open at –

http://localhost:3000

Phase 1 –

Phase 2 (AI-Enhanced) –
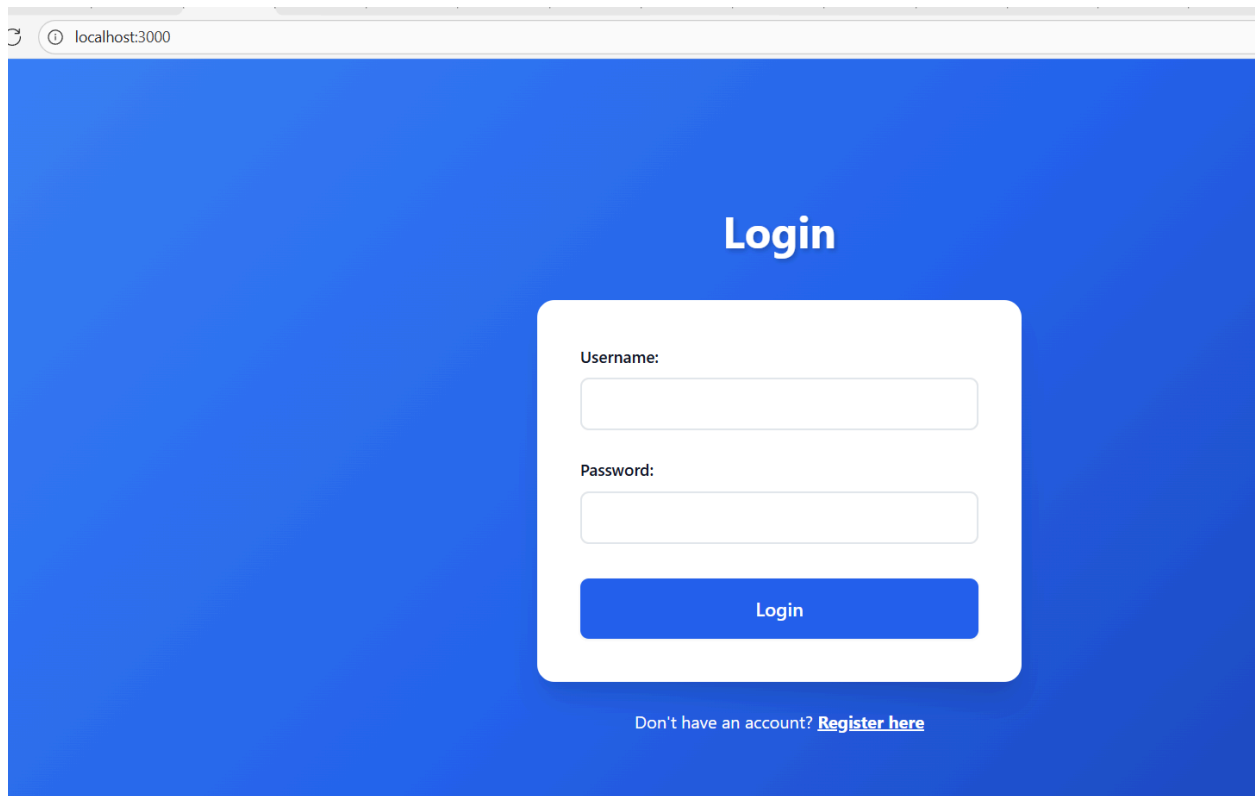
Backend Setup:

Navigate to the backend/ScholarLendingPortal

Install `sudo apt install python3.12-venv`

Create Virtual env -

`python3 -m venv venv`

`source venv/bin/activate`

Install dependencies -

`pip install -r requirements.txt`

Install setup tools -

`pip install setuptools`

Run this -

```
python manage.py makemigrations api
```

Run Migration -

```
python manage.py migrate
```

If you get while running migration, then follow below steps,

```
rm db.sqlite3
```

```
rm api/migrations/0*.py
```

ls api/migrations – should see **__init__.py**

Recreate fresh migrations,

```
python manage.py makemigrations
```
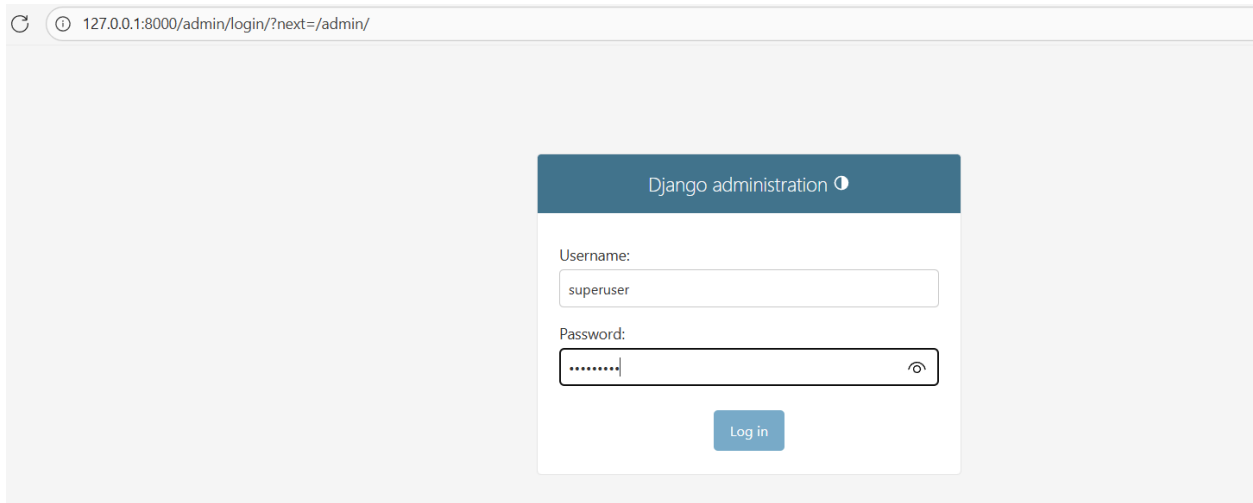
```
python manage.py migrate
```


Create a superuser --

```
python manage.py createsuperuser
```
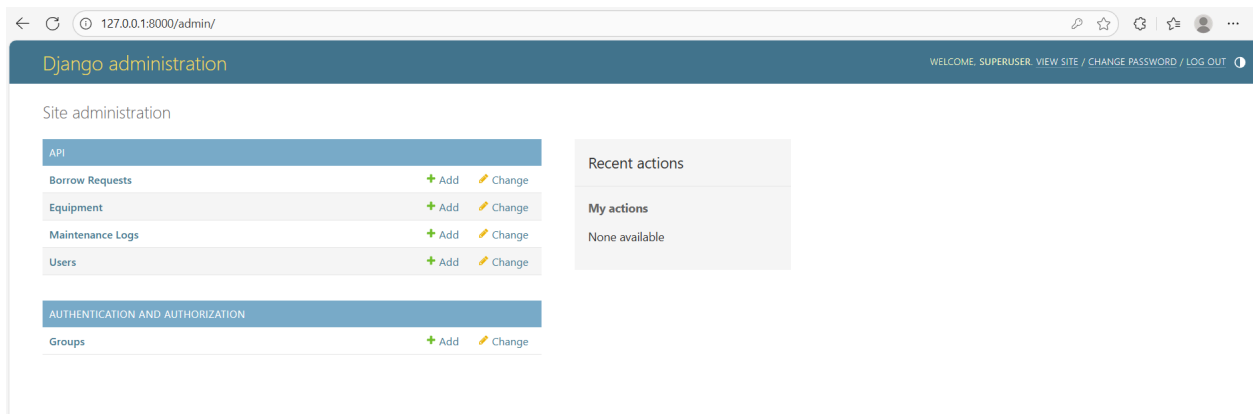
Start the server -

```
python manage.py runserver
```

Once backend server is started, will run at http://127.0.0.1:8000

Access the Admin Panel - Visit http://127.0.0.1:8000/admin/ and login with your superuser credentials.

Once logged in using superuser creds, below dashboard will appear,

Now to create new user, go to http://localhost:3000

Click on **Register Here**,

Phase 1 –

localhost:3000/register

Register

Username: [   ]
Email: [   ]
Password: [   ]
First Name: [   ]
Last Name: [   ]
Role: Student ▾
Phone Number: [   ]
[Register]

Already have an account? Login here

Phase 2 (AI Enhanced) –

localhost:3000/register

Username:

[   ]

Email:

[   ]

Password:

[   ]

First Name:

[   ]

Last Name:

[   ]

Role:

[Student   ▾]

Phone Number:

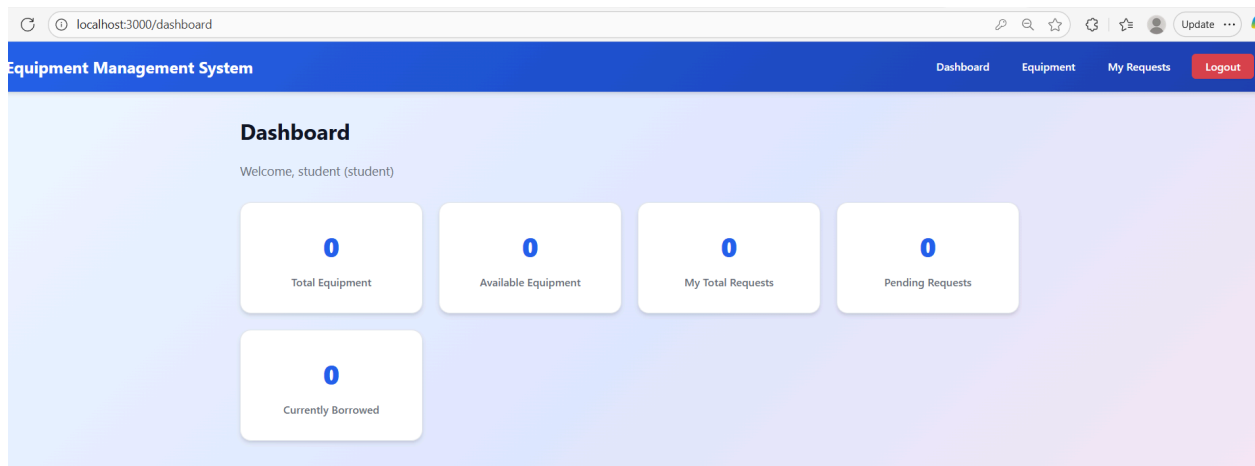[   ]

[Register]

Create 3 users by selecting role.

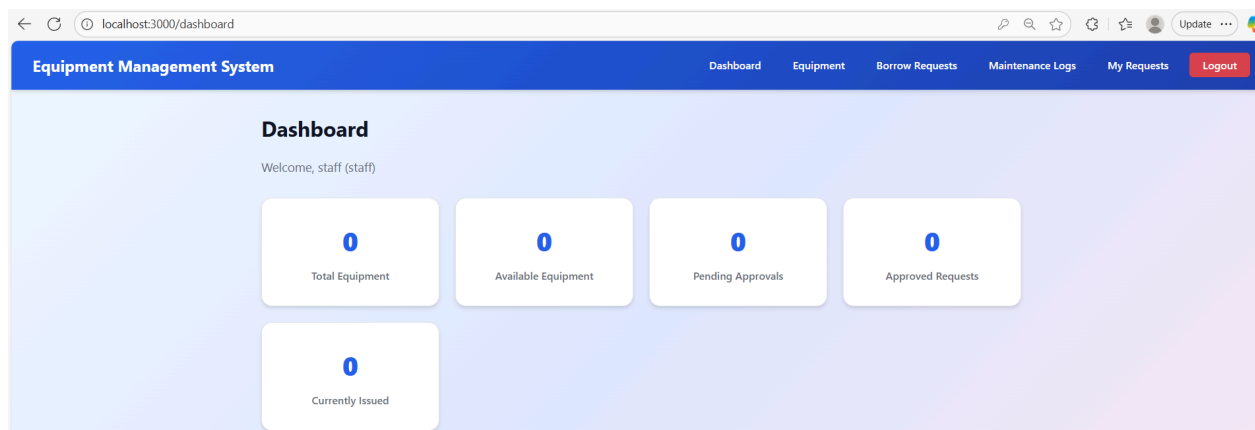**student** (who will request / borrow the equipment)

staff (who will reject / approve the request)

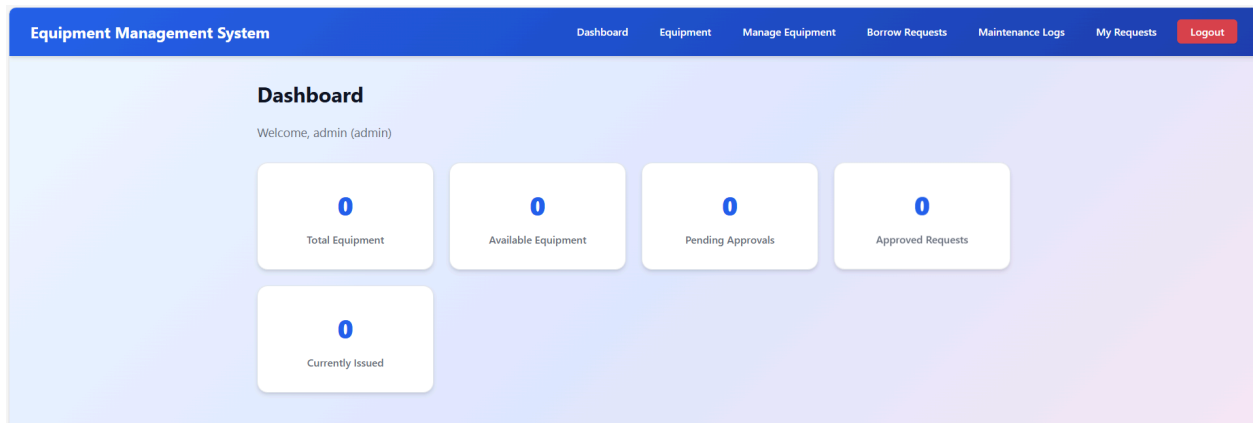**admin** (who will take care of addition or removal of equipments and other stuffs)
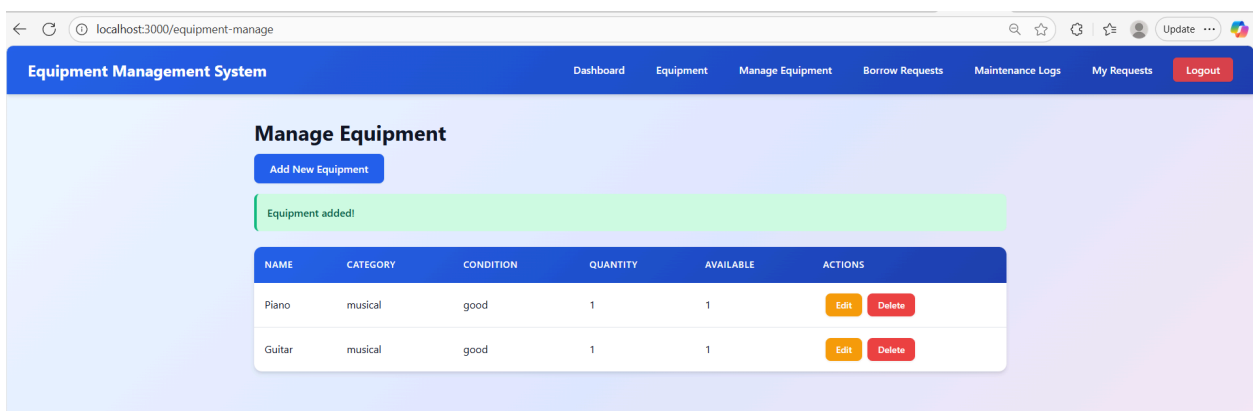
Student Dashboard (Phase 2) –

Staff Dashboard (Phase 2) –

Admin Dashboard (Phase 2) –

Log in as Admin and add equipment by going to **Manage Equipment** section and click on **Add equipment**,

Once Equipment is available, Student will borrow/lend the equipment for specific period of time. Log in as Student, equipment will be available to borrow.

To Borrow Equipment, go to **Equipment** section and click on **Borrow** for the equipment which needs to be borrowed.
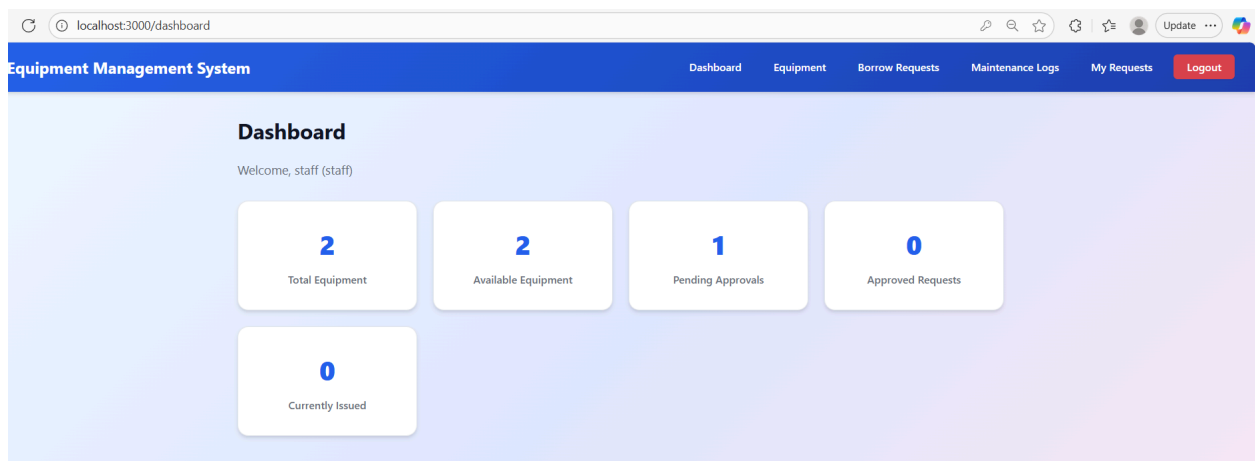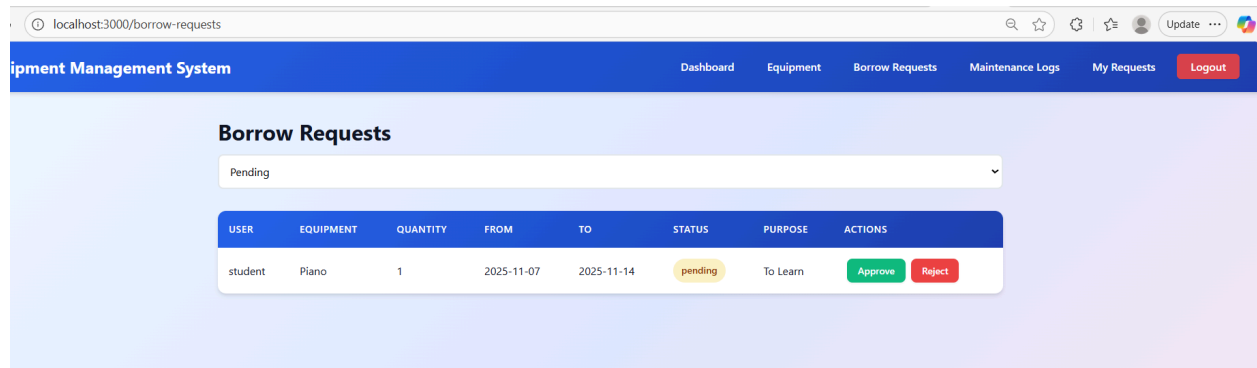
Provide essential details to borrow the equipment and click on **Submit Request**.

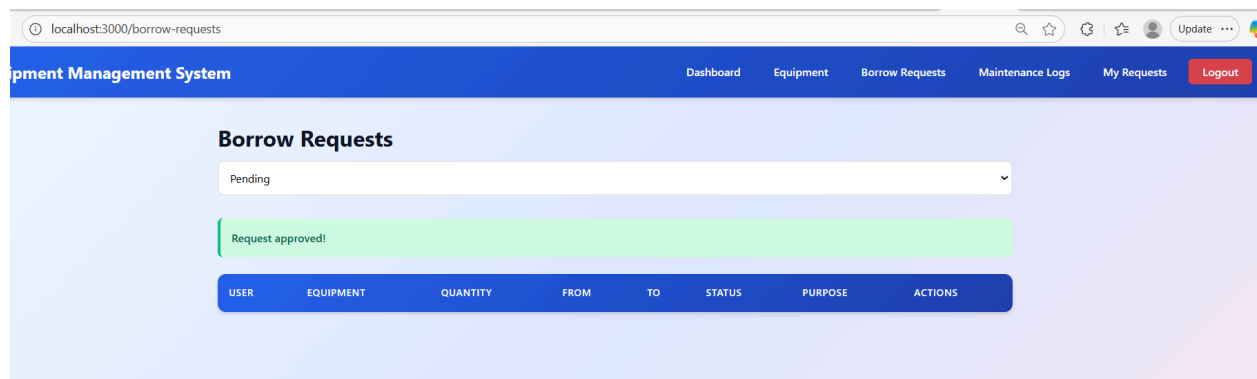Staff will approve/reject the request. Log in as Staff.

There is one pending request approval, raised by student.

Now to accept the request, go to **Borrow Request** and approve it.

Once request is approved, Equipment becomes available to Student for specific period of time.

Login as Student and go to **My Requests** section, Request status will be **approved.**