

# Virtual logbook and course hours tracking

## Contents

1. Introduction.....	2
2. Technologies Used.....	3
3. Features.....	3
4. System Architecture.....	3
5. Business Logic.....	4
6. Implementation Details.....	5
7. User Interface.....	5
8. Running the Application.....	11
8.1. Running the Django Backend.....	11
8.2. Running the React Frontend.....	12
9. Conclusion.....	13

# Virtual logbook and course hours tracking

## Virtual logbook and Course hours tracking

### 1. Introduction

The Virtual Logbook and Course Hours Tracking web application is designed to help users keep track of different topics taught during courses and the number of hours spent on these courses. The application is structured to support both regular users and class delegates/prefects. Here's a detailed overview of the system's functionalities, architecture, and instructions on how to set up and run the application.

#### Key Features

- **User Roles:** The application supports two types of users:
  - **Regular Users:** These users belong to a particular class and can view logbook entries for their courses.
  - **Class Delegates/Prefects:** These users also belong to a particular class but have additional responsibilities, including filling out the logbook and tracking course hours.
- **Logbook Entries:** Class delegates/prefects can create logbook entries that include:
  - **Subject:** The course for which the entry is being made.
  - **Date:** The specific day the entry pertains to.
  - **Period:** The start and end times of the teaching session.
  - **Teacher:** The teacher who conducted the session.
  - **Text:** Detailed notes or descriptions of the session.
  - **Attachments:** One or more files (images, audio, etc.) related to the session.
- **Automatic Hour Logging:** The application automatically logs the number of hours a teacher has taught a particular course whenever a logbook entry is created.
- **Course Views:** Regular users can view a list of courses and see all logbook entries associated with a selected course.

# Virtual logbook and course hours tracking

## 2. Technologies Used

- Django: Used as the backend framework for building the web application.
- React: Employed for developing the frontend user interface to enhance interactivity.
- PostgreSQL: Chosen as the database management system for storing data reliably and efficiently.
- React Query: Utilized for managing server-state and handling data fetching in the frontend.
- HTML/CSS: Used for structuring the user interface and styling components.
- Git/GitHub: Employed for version control and collaborative development.

## 3. Features

- User Authentication: Allows users to register, login, and manage their accounts securely.
- Logbook Entry Creation: Enables teachers to create logbook entries with details such as start time, end time, course, and text.
- Course Management: Provides administrators with the ability to add, edit, and delete courses.
- Automatic Hour Tracking: Automatically calculates teaching hours based on logbook entries and updates course hour totals.
- File Upload: Allows users to upload files and associate them with logbook entries for documentation purposes.

## 4. System Architecture

The system follows a client-server architecture, with the backend implemented using Django and the frontend developed using React. Data is stored in a PostgreSQL database, with Django ORM handling database interactions. The frontend communicates with the backend via RESTful APIs, facilitating seamless data exchange.

# Virtual logbook and course hours tracking

## 5. Business Logic

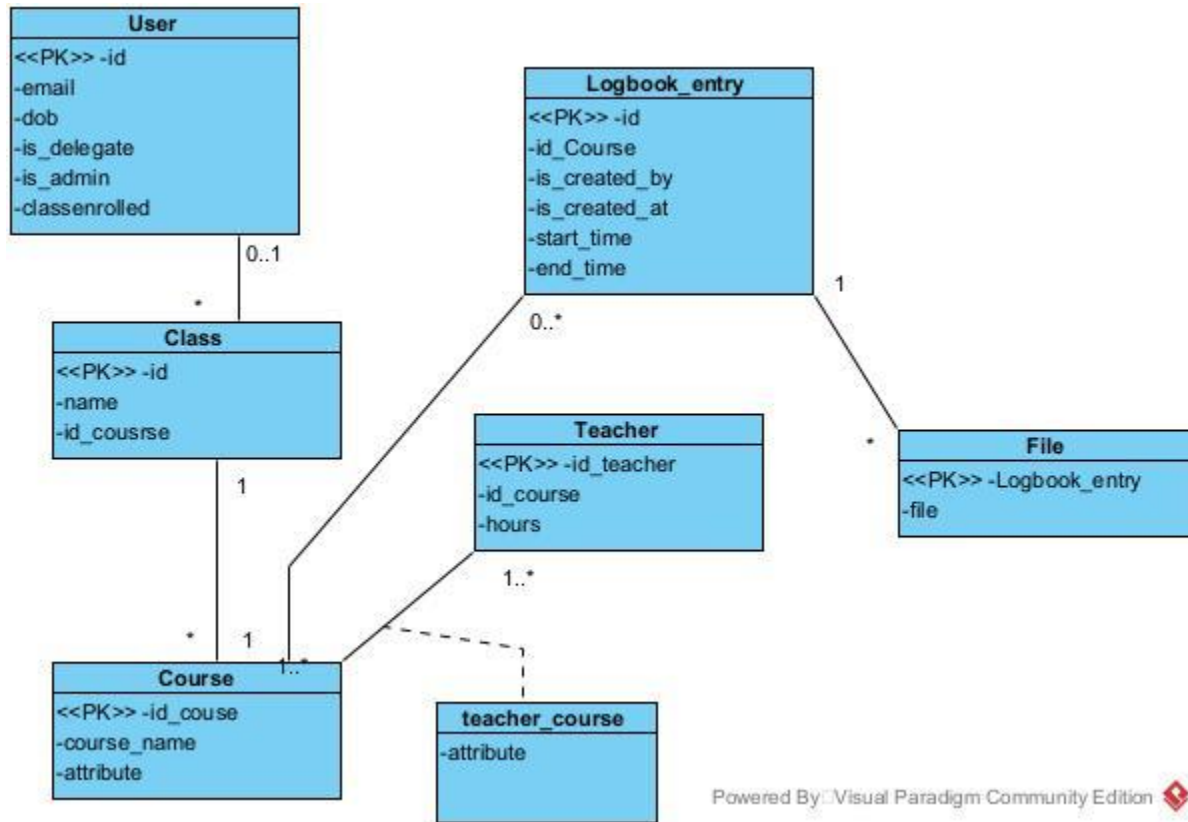


Figure 1: Class diagram

### User Class:

- Represents the user model with attributes like email, name, date\_of\_birth, gender, etc.
- It has relationships like is\_admin and is\_delegate, which determine the user's permissions.
- Also has a foreign key relationship with the Class model, indicating the class the user is enrolled in.

### Course Class:

- Represents a course with attributes like name.
- This class doesn't have any explicit relationships in the diagram, but it's implied that it may have many logbook entries associated with it.

### Class Class:

- Represents a class with attributes like name.
- It has a many-to-many relationship with the Course model, indicating that a class can have multiple courses, and a course can belong to multiple classes.

# Virtual logbook and course hours tracking

## Teacher Class:

- Represents a teacher with attributes like name.
- It doesn't have any explicit relationships in the diagram, but it's implied that a teacher can have many logbook entries associated with them.

## LogbookEntry Class:

- Represents an entry in the logbook with attributes like start\_time, end\_time, text, etc.
- It has foreign key relationships with the Course, Teacher, and User models, indicating the course, teacher, and user associated with the logbook entry.

## File Class:

- Represents a file associated with a logbook entry.
- It has a foreign key relationship with the LogbookEntry model, indicating the logbook entry the file belongs to.

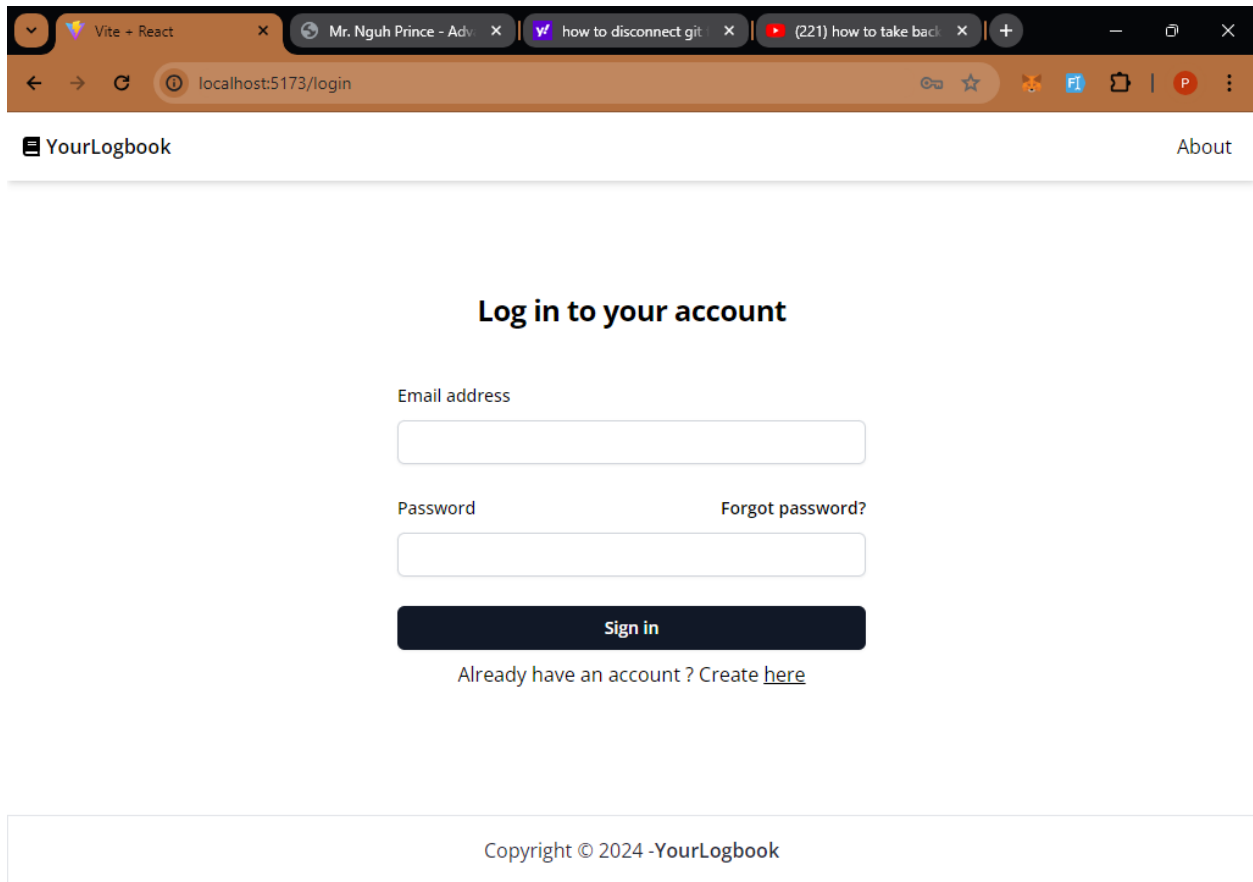
## 6. Implementation Details

The implementation involved building Django models for users, courses, logbook entries, and file uploads. React components were created for the user interface, with React Query used for managing state and data fetching. The system's logic for calculating teaching hours was implemented in the Django backend, ensuring accuracy and consistency.

## 7. User Interface

The user interface features an intuitive design with clear navigation and interactive elements. The delegate can easily log the teaching hours of teachers, while administrators have access to tools for managing courses and monitoring student activities. The interface is responsive, allowing users to access the system from various devices.

# Virtual logbook and course hours tracking



The screenshot shows a web browser window with multiple tabs. The active tab is titled 'localhost:5173/login'. The browser's address bar shows the URL 'localhost:5173/login'. The page header includes the text 'YourLogbook' on the left and 'About' on the right. The main content area features a login form with the heading 'Log in to your account'. The form contains two input fields: 'Email address' and 'Password'. To the right of the password field is a link labeled 'Forgot password?'. Below the password field is a dark blue button with the text 'Sign in'. At the bottom of the form, there is a text link: 'Already have an account ? Create [here](#)'. The footer of the page contains the text 'Copyright © 2024 -YourLogbook'.

Vite + React x Mr. Nguh Prince - Adv. x how to disconnect git x (221) how to take back x

localhost:5173/login

YourLogbook About

## Log in to your account

Email address

Password [Forgot password?](#)

**Sign in**

Already have an account ? Create [here](#)

Copyright © 2024 -YourLogbook

Figure 2: Login screen

# Virtual logbook and course hours tracking

YourLogbook

About

Create your account

Full Name

Email address

Date of Birth

mm/dd/yyyy

Gender

Select Gender

Class enrolled

Select your class

Password

Create account

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

Figure 3: Registration page

# Virtual logbook and course hours tracking

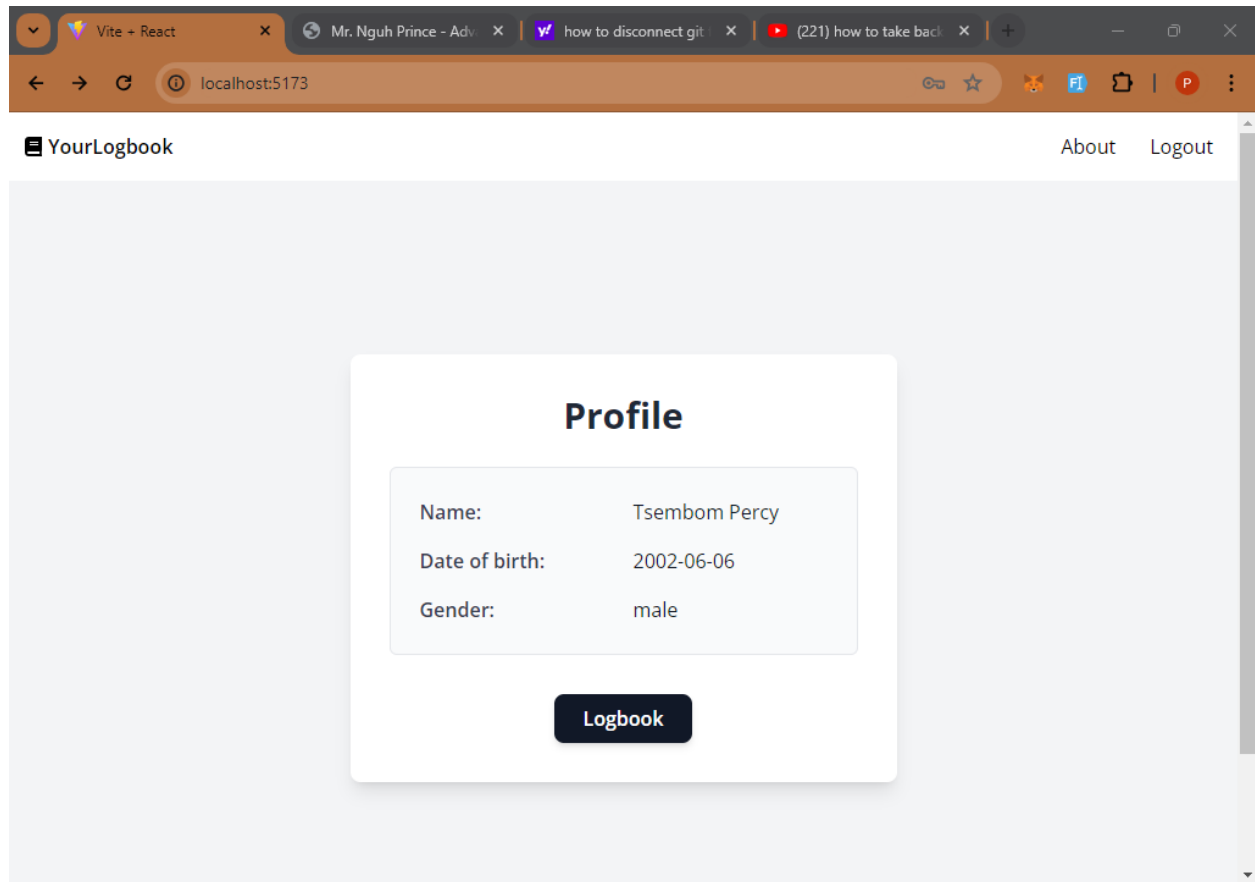


Figure 4: Home page



# Virtual logbook and course hours tracking



Figure 5: Select course

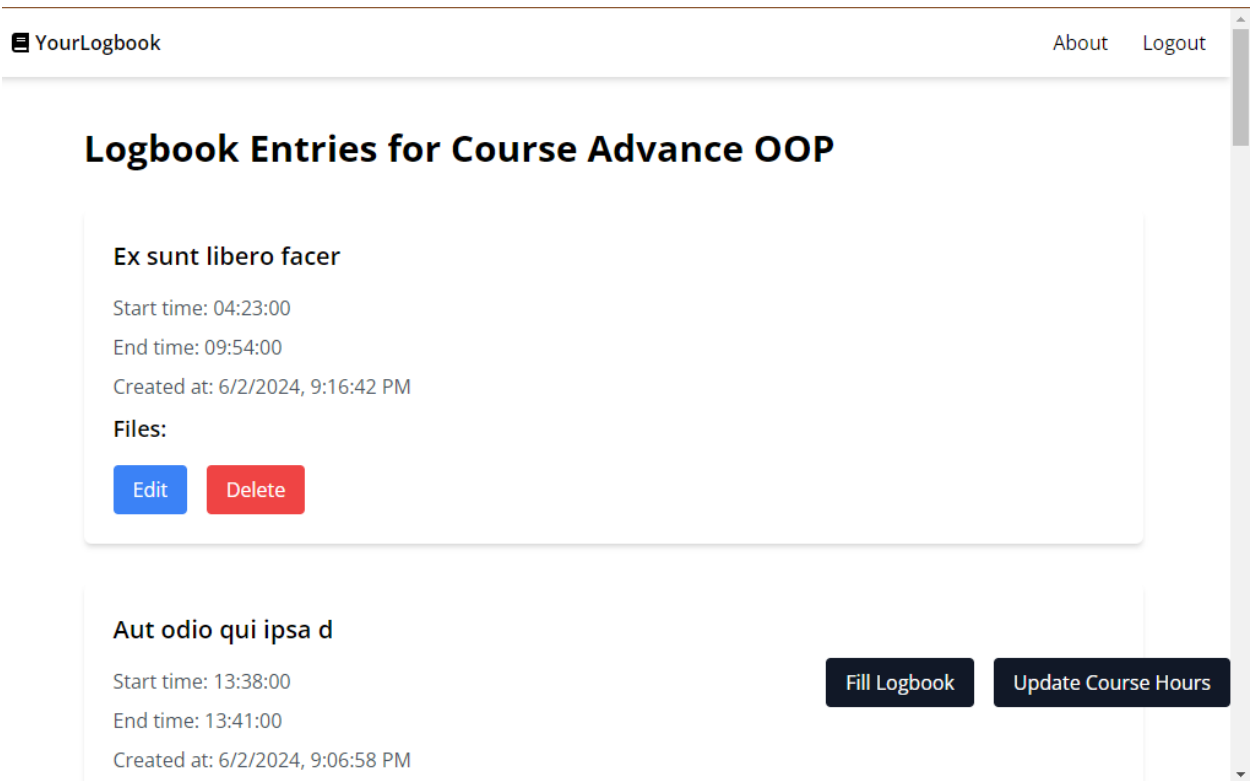


Figure 6: Logbook entries

# Virtual logbook and course hours tracking

The screenshot shows the 'YourLogbook' application interface. A modal titled 'New Logbook Entry' is open, allowing users to create a new logbook entry. The modal contains the following fields and controls:

- Start Time:** A time selection field showing '--:-- --' with a clock icon.
- End Time:** A time selection field showing '--:-- --' with a clock icon.
- Text:** A large text area for entering details.
- Teacher:** A dropdown menu with the placeholder text 'Select Teacher'.
- Upload Files:** A button labeled 'Choose Files' and a status indicator 'No file chosen'.
- Buttons:** 'Cancel' and 'Save' buttons at the bottom of the modal.

In the background, the 'Logbook Entries' section is visible, showing two entries with details like 'Ex sunt libero facer' and 'Aut odio qui ipsa d', including their start/end times and creation dates. At the bottom right, there are buttons for 'Fill Logbook' and 'Update Course Hours'.

Figure 7: Fill Logbook form

The screenshot shows the 'YourLogbook' application interface with the 'Logbook Entries for Course Advance OOP' section. A modal titled 'Hours Taught' is open, allowing users to update the hours taught for a specific entry. The modal contains the following fields and controls:

- Name:** A text field displaying 'Mr. Nguh Prince'.
- Hours:** A text field displaying '20.00'.
- Teacher:** A dropdown menu with the placeholder text 'Select Teacher'.
- Hours Taught:** A text input field for updating the hours.
- Buttons:** 'Cancel' and 'Save' buttons at the bottom of the modal.

In the background, the 'Logbook Entries' section is visible, showing two entries with details like 'Ex sunt libero facer' and 'Aut odio qui ipsa d', including their start/end times and creation dates. At the bottom right, there are buttons for 'Fill Logbook' and 'Update Course Hours'.

Figure 8: Update hours taught

# Virtual logbook and course hours tracking

## 8. Running the Application

### 8.1. Running the Django Backend

#### 1. Setup Virtual Environment:

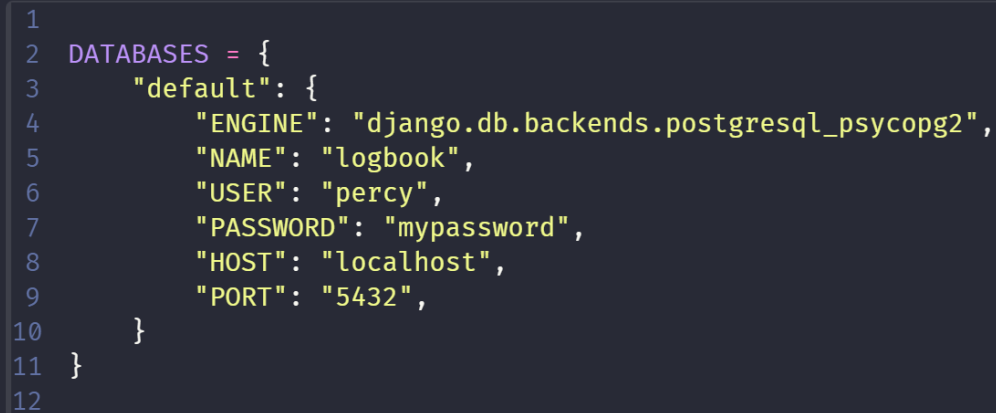
- Navigate to the root directory of the Django project.
- Create a virtual environment using `python -m venv myenv`
- Activate the virtual environment using `.myenv\Scripts\activate`

#### 2. Install Dependencies

Run `pip install -r requirements.txt`

#### 3. Database Setup:

- Configure PostgreSQL database settings in `settings.py`.



```
1
2 DATABASES = {
3     "default": {
4         "ENGINE": "django.db.backends.postgresql_psycopg2",
5         "NAME": "logbook",
6         "USER": "percy",
7         "PASSWORD": "mypassword",
8         "HOST": "localhost",
9         "PORT": "5432",
10    }
11 }
12
```

Figure 9: Database settings

# Virtual logbook and course hours tracking

Make sure your database should respect the above.

- Run migrations to create database schema:

``

```
python manage.py makemigrations
```

```
python manage.py migrate
```

``

- After that run `python manage.py restore_db` to restore the database.

## 4. Run Development Server

Type `python manage.py runserver`

The Django development server should now be running locally at `http://localhost:8000`.

### 8.2. Running the React Frontend

#### 1. Navigate to Frontend Directory:

Type `cd frontend`

#### 2. Install Dependencies:

Type `npm install`

#### 3. Start Development Server:

Type `npm start`

The React development server should now be running locally at `http://localhost:3000`.

## Accessing the Application

Once both the Django backend and React frontend servers are running, you can access the application by opening your web browser and navigating to `http://localhost:5173`. This will load the React frontend, which communicates with the Django backend API running at `http://localhost:8000`.

# **Virtual logbook and course hours tracking**

## **9. Conclusion**

The "Virtual Logbook and Course Hour Tracking" project has successfully addressed the need for an efficient and reliable system for logging teaching hours and tracking course hours. By leveraging modern technologies and best practices, the system provides a user-friendly platform that enhances productivity and accuracy in educational settings.