Name -Muskan Kumari, Bits Id -2023tm93675

**School Vaccination Portal**

**1. System Overview**

The School Vaccination Portal is a React + Flask/PostgreSQL web-based application designed to streamline the planning, scheduling, and management of vaccination drives within educational institutions. It allows administrators to create vaccination drives, assign students to these drives, and monitor their vaccination status effectively.

**Key Objectives:**

* Simplify the organization of vaccination drives.
* Maintain accurate records of student vaccinations.
* Improve communication between administrators and participants.
* Ensure compliance with health guidelines and deadlines.

**Key Features:**

* Student Management: View, add, and manage pupil records.
* Vaccination Drive Management: Create and schedule vaccination drives.
* Drive-Studennt Assignment: Assign students to specific drives based on availability or requirement.
* Status Tracking: Monitor and update vaccination statuses.
* API Access: RESTful API endpoints for all major operations.

**2. Technology Stack:**

* Frontend: React.js (with Bootstrap for styling)
* Backend: Python Flask (API and business logic)
* Database: PostgreSQL (data storage and management)
* API Testing/Docs: Postman or Swagger
* Authentication: Hardcoded login
* API Documentation: Postman Collection
* Version Control: Git & GitHub

**3. Application Architecture**

The application follows a three-tier architecture:

* Frontend: React.js
  + Provides user interface for interacting with the system.
  + Communicates with backend via RESTful APIs.
* Backend: Flask (Python)
  + Handles API logic, validation, and database interactions.
* Database: PostgreSQL
  + Stores data related to pupils, vaccination drives, and vaccination records.

**3.1 High-Level Architecture**

Frontend (React)  
 |  
 | REST API calls  
 ↓  
Backend (Flask)  
 |  
 | Query builder  
 ↓  
Database (PostgreSQL)

**3.2 Modules**

- Authentication Module  
- Dashboard Module  
- Student Management  
- Vaccination Drive Management  
- Reporting Module

**4. Database Schema (Sample Tables/Collections)**

4.1 Student Table

|  |  |  |
| --- | --- | --- |
| Field Name | Type | Description |
| id | UUID / ObjectId | Primary Key |
| name | String | Student name |
| student\_class | String | Class |
| vaccination\_status | String | Vaccination Status |
| vaccination\_date | Date | Date of vaccination |
| vaccine\_name | String | Vaccine Name |

4.2 vaccination\_drive

|  |  |  |
| --- | --- | --- |
| Field Name | Type | Description |
| id | UUID / ObjectId | Primary Key |
| vaccine\_name | String | Name of vaccine |
| vaccination\_date | Date | Date of drive |
| available\_doses | Number | No. of doses available |
| applicable\_classes | String | Class Eligible |

**5. Frontend-Backend Interaction**

| Frontend Page | Backend Endpoint | Method | Functionality |
| --- | --- | --- | --- |
| /login | *none* | - | Simulated login |
| /dashboard | /api/metrics | GET | Dashboard stats |
| /manage | /api/students | GET | List students |
|  | /api/studentdata | POST | Add student |
|  | /api/vaccination\_upload | POST | Upload CSV |
| /drives | /api/drives | GET | List drives |
|  | /api/adddrives | POST | Create drive |
|  | /api/drives/<id> | PUT | Edit drive |
| /reports | /api/reports/vaccinations | GET | Filtered reports |

**6. API Documentation**

Here’s a sample document in Postman

GET /api/metrics

{

"total\_students": 20,

"vaccinated\_count": 10,

"vaccinated\_percent": 50.00,

"upcoming\_drives": [

{ "name": "Hepatitis A", "date": "2025-06-01" }

]}

POST /api/studentdata

CopyEdit

// Body

{ "name": "John Doe",

"class": "5",

"vaccination\_status": "yes",

"vaccine\_name": "Covaxin",

"vaccination\_date": "2025-05-05"

}

// Response

{ "message": "Student added" }

POST /api/adddrives

{

"vaccine\_name": "Polio",

"drive\_date": "2025-06-01",

"available\_doses": 50,

"applicable\_classes": "1,2,3"

}

**7. Assumptions/ Business Logic & Validations**

* Only one school admin logs in using hardcoded credentials (Muskan, Muskan1402)
* CSV upload works using form-data with a .csv file
* A student cannot receive the same vaccine twice
* Past vaccination drives are not editable
* Drives must be created ≥15 days in advance
* Applicable classes are stored as string ("5,6,7")

**8. UI/UX Wireframes**

* Login Page: Login with username/password
* Home Page: Dashboard with tabs for Total Student, Vaccinated Students and Upcoming Drives.
* Student Management Page: Table of Student with CRUD actions.
* Drives Page: Calendar view and list of upcoming drives.
* Vaccination Page: Form to record vaccination for Students.

**9. Run Instructions**

**Backend**

* Requirements: Python 3.12, pip, Flask, SQLAlchemy
* Database: VaccPortal

User: postgres

Password: Muskan\_1402

* Install dependencies: pip install -r requirements.txt
* Run: python app.py

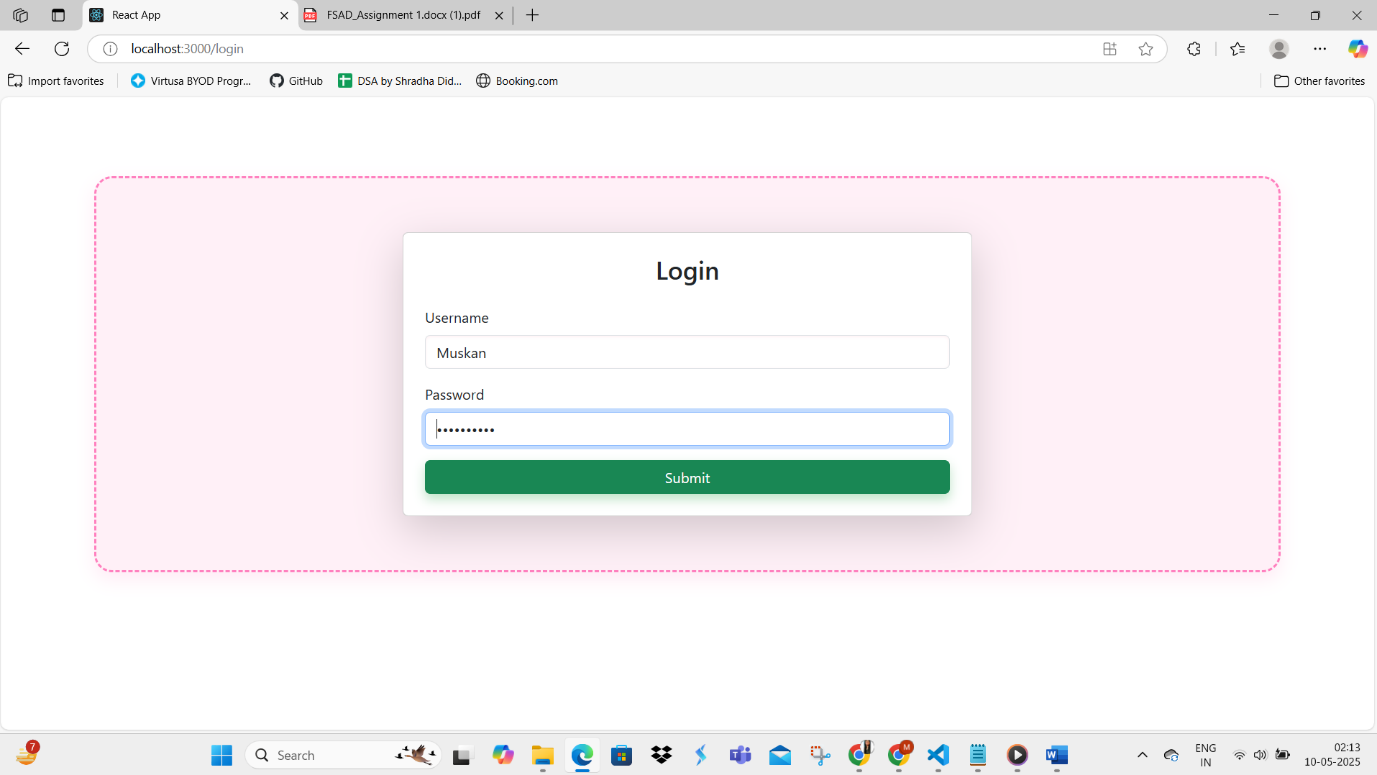
**Frontend**

* Requirements: Node.js, npm
* Install dependencies: npm install
* Run: npm start

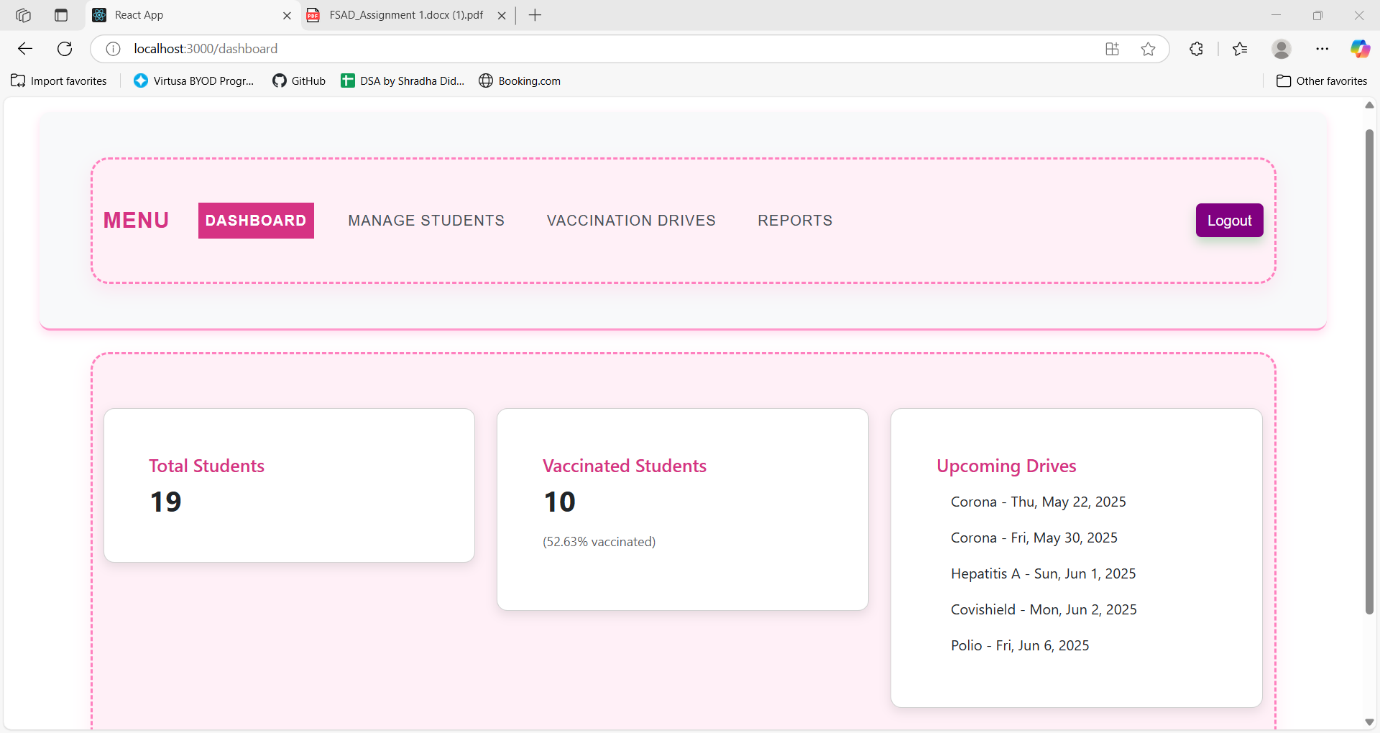
**10. Output Snapshots**

**UI Snapshots**

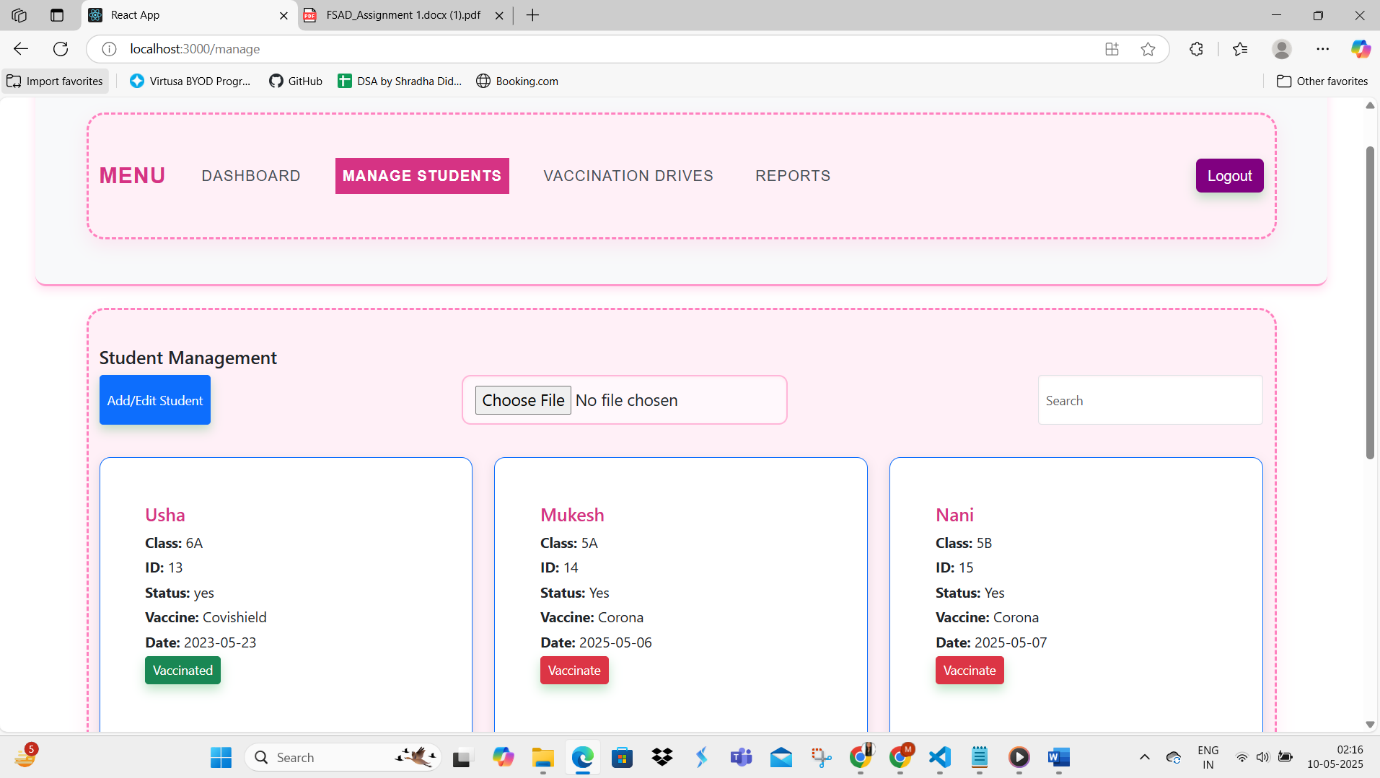
* Login page with Username: Muskan, Password: Muskan1402



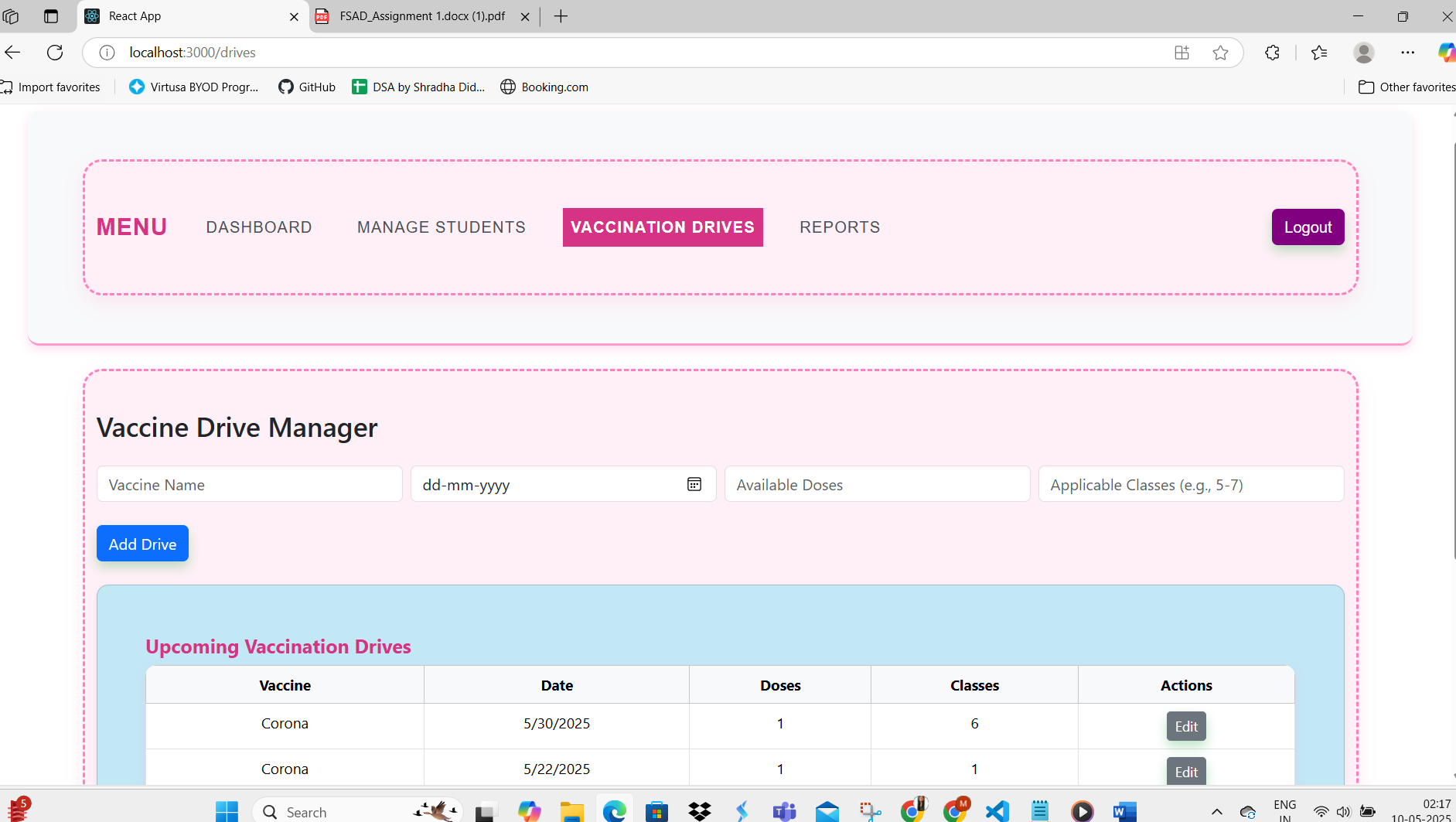
* Dashboard: Total students, vaccinated %, upcoming drives



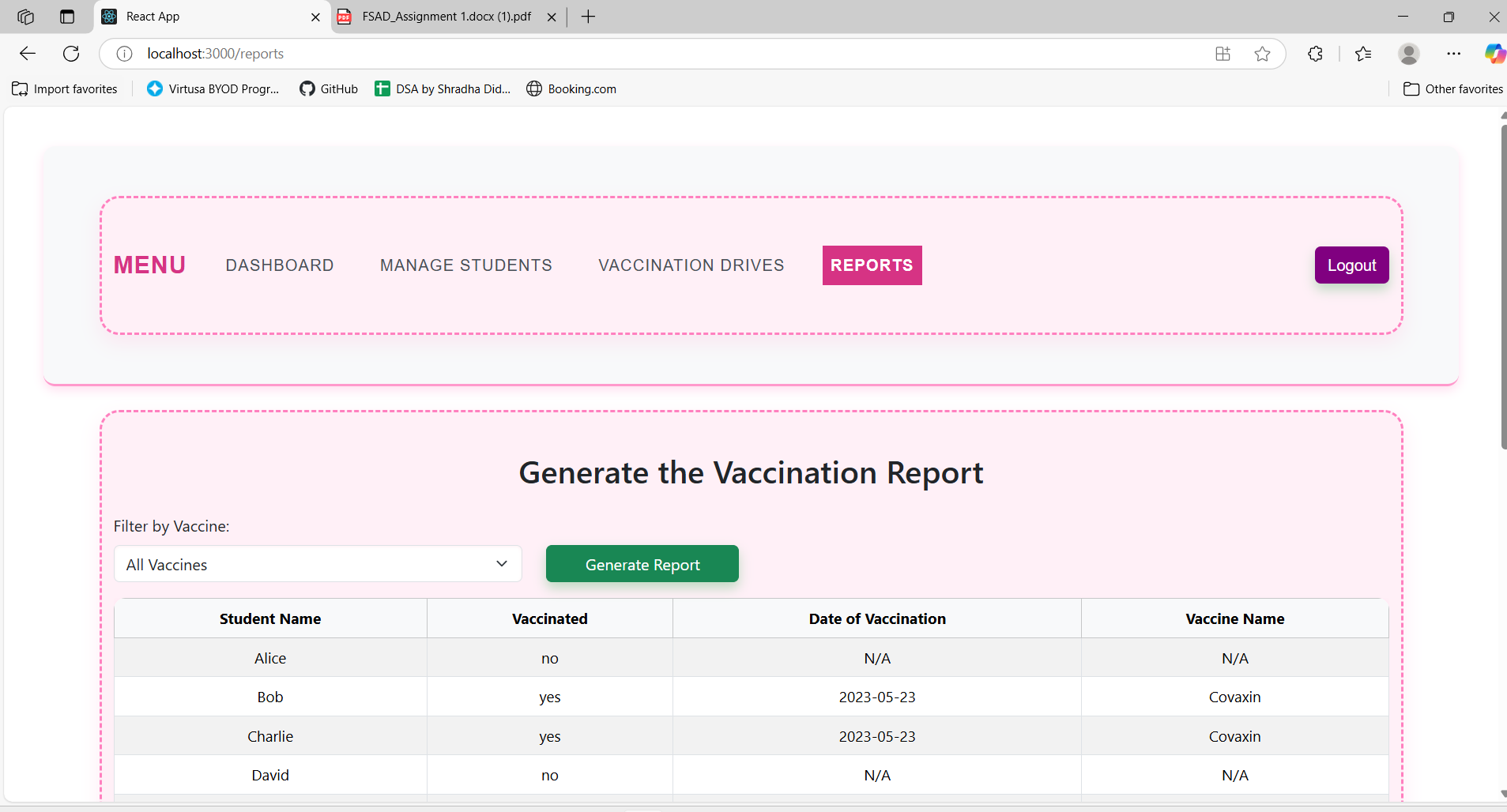
* Student table and Add/Edit modal

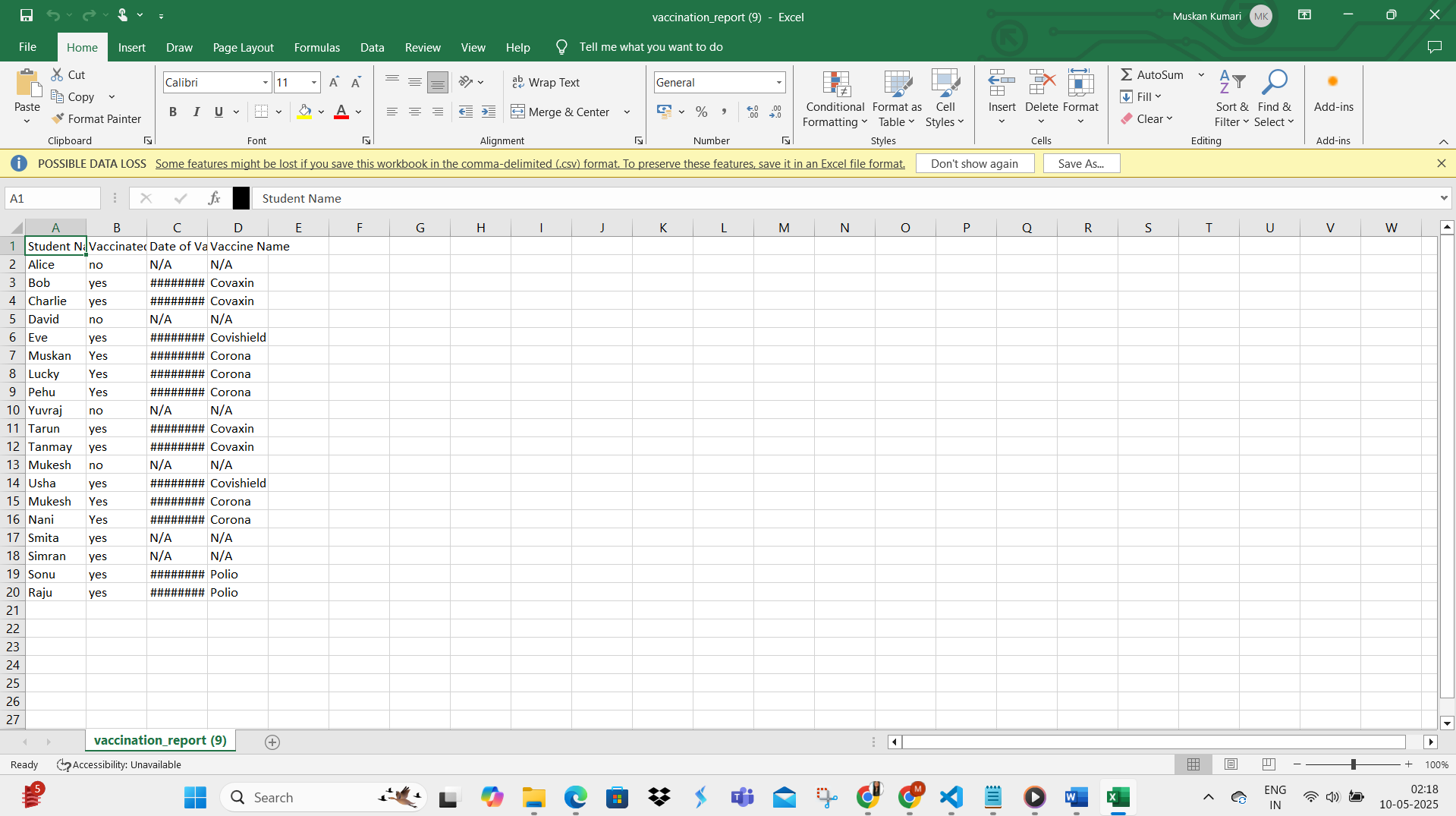


* Vaccination Drive table + form

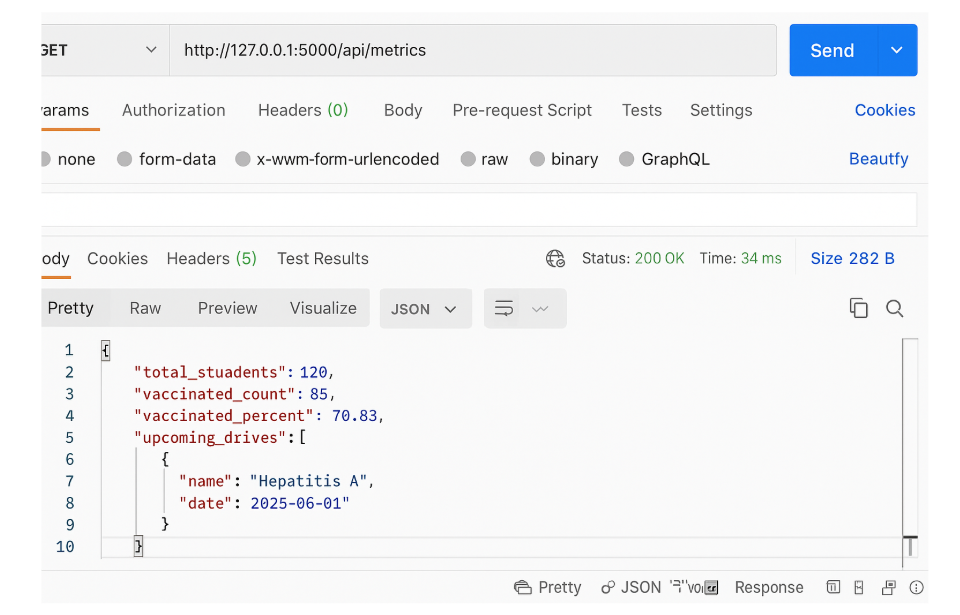


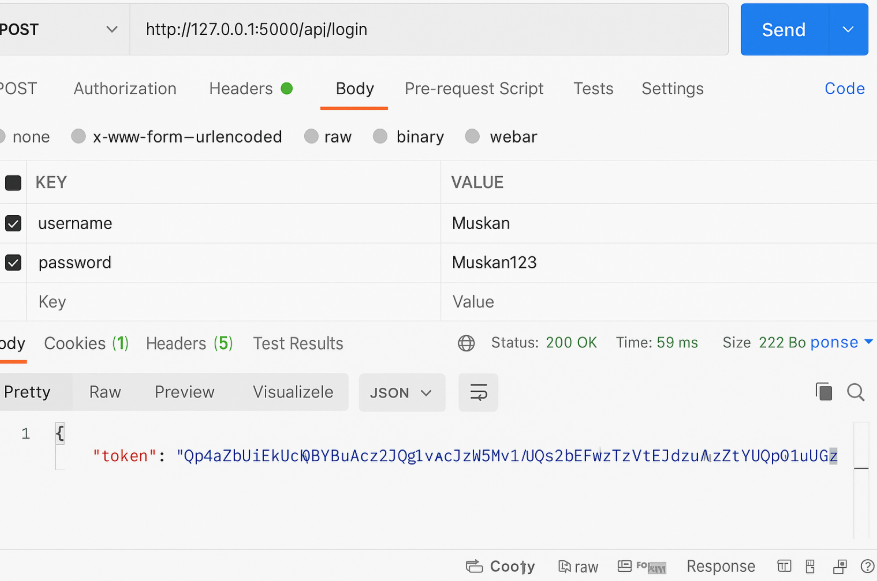
* Report table with export to CSV





**API (Postman) Snapshots**





**11. GitHub and Demo Video**

**GitHub Repo:** https://github.com/muskan1402/StudentVaccinationPortalBE/tree/master

**Demo Video Link:**

https://drive.google.com/uc?id=1E-FK8shnFB7d8UMkjwOlDQ9y5b5IfOX4&export=download

https://1drv.ms/v/c/a121f060944a9224/ERIRbHUIQhRFoxFk6kwLOAAB8aSPbLNY\_mZL\_bB6YwpgbA