

FSAD Assignment 1

Mallidi Akhil Reddy

2024TM93056

Vaccination Management System Documentation

System Overview:

This Vaccination Management System is developed to manage student vaccinations in a school. It provides functionalities to:

- Manage student records.
- Schedule and track vaccination drives.
- Edit Vaccination drives.
- Display key metrics (total students, vaccinated students, upcoming drives).
- Download student records as files, along with application of filters.

Full Application Architecture:

- **Frontend:** React, Sass.
- **Backend:** Express.js, Node.js.
- **Database:** MongoDB.
- **API Documentation:** Postman.
- **Deployment:** Any cloud provider with a Node.js setup.

Architecture Diagram:



Frontend-Backend Interaction:

- Frontend along with store thunk action sends API requests to the backend using Axios
- Backend responds with JSON data and appropriate status code, handling error scenarios as well.
- API endpoints are secured and protected, following RESTful principles.

API Endpoints:

- **POST /api/users/login:** Authenticates the user and generate a JWT token.
- **GET /api/students:** Fetches all students across all the classes.
- **POST /api/students:** Creates a new student data row with the data passed.

- **PUT /api/students/:id:** Updates a student record having the mentioned id.
- **POST /api/students/bulkupload:** Creates multiple student records based on the CSV file uploaded.
- **GET /api/vaccine:** Fetches all vaccines available in the DB.
- **POST /api/vaccine:** Creates a new vaccine record.
- **POST /api/drive:** Creates a new drive.
- **GET /api/drive:** Fetches all drives available.
- **GET /api/drive/status:** Fetches aggregate data combining students drives and vaccine models.
- **PATCH /api/drive/vaccinate:** Records student as vaccinated and a dose usage in drive model.
- **GET /api/drive/:id:** Fetches a drive data with that id.
- **PUT /api/drive/:id:** Updates the drive data with that id.
- **GET /api/dashboard/metrics:** Fetches dashboard metrics (total students, vaccinated students, upcoming drives).

API Documentation:

API documentation has been published and can be viewed in the below link –

<https://documenter.getpostman.com/view/6868293/2sB2jAa7hR>

Database Schema:

Collection: Drive

_id: ObjectId
name: String
vaccine: ObjectId
date: Date
numDoses: Number
usedDoses: Number
applicableClasses: String
createdBy: ObjectId
createdAt: Date
updatedAt: Date

Relationships:

vaccine - Vaccine Model (_id)
createdBy - User Model (_id)

Collection: Student

_id: ObjectId
name: String
regNo: String

standard: String
dob: String
parentName: String
parentContactNum: String
vaccinations: Array of objects

Collection: User

_id: ObjectId
firstName: String
lastname: String
fullName: String
email: String
password: String
isAdmin: Boolean

Collection: Vaccine

_id: ObjectId
name: String
manufacturer: String

Assumptions

- Only students in applicable classes will be shown in the drive.

- A student can receive different vaccinations, which each vaccine limited to once.
- Drives are displayed in the dashboard only if they are scheduled for the next 30 days.
- Only eligible students (filtered out students who has received that vaccine previously in applicable class) are shown in drive.
- As the portal is specific to each school, admin user can be added manually from Backend.

Deliverables:

FE Code – <https://github.com/itzzmeakhi/vaccinate-portal-fe>

BE Code – <https://github.com/itzzmeakhi/vaccinate-portal-be>

Video -

https://drive.google.com/file/d/1JXUxkp8Pq9ENU7SDbGXuNzb3_nCn_kb1/view

Thank you