# Vaccination Management System  Design Document

## Overview

This system is designed to facilitate the management of vaccination drives within a school environment. It allows school coordinators to:

- Manage student records.

- Organize vaccination drives.

- Track vaccination statuses.

- Generate comprehensive reports.

## Architecture

- Frontend: React.js with Ant Design (antd) components.

- Backend: Node.js with Express.js.

- Database: MongoDB with Mongoose ODM.

## Project Structure

masters-full-stack-assignment/

 backend/

    controllers/

    models/

    routes/

    server.js

 frontend/

    components/

    pages/

    App.js

 README.md

## Data Models

### 1. Student
- id: String (unique)
- name: String

- class: String

- vaccination_status: Boolean

- vaccination_drives: [String] (Array of vaccination drive names)

## 2. VaccinationDrive

- name: String (unique)

- date: Date

- dosesCount: Number

- applicableClasses: [String]

## API Endpoints

### 1. Student Management

- POST /students: Add a new student.

- GET /students: Retrieve all students.

- PUT /students/:id: Update student details.

- DELETE /students/:id: Remove a student.

### 2. Vaccination Drive Management

- POST /vaccination-drives: Create a new drive.

- GET /vaccination-drives: Retrieve all drives.

- PUT /vaccination-drives/:name: Update drive details.

- DELETE /vaccination-drives/:name: Remove a drive.

### 3. Vaccination Status

- PUT /students/:id/vaccination-status: Update a student's vaccination status.

### 4. Reports

- GET /reports/vaccination: Generate vaccination reports with optional filters.

## Frontend Components

- StudentList: Displays all students.

- VaccinationDriveList: Displays all vaccination drives.

- VaccinationStatusForm: Form to update a student's vaccination status.

- ReportGenerator: Interface to generate and view reports.

## Reporting Features

- Filters: By vaccination name.

- Details: Student name, class, vaccination status, vaccine name, date.

- Export Options: CSV, Excel, PDF (to be implemented).

## Acceptance Criteria

- Student Management: Ability to add, view, edit, and delete student records.

- Vaccination Drives: Ability to create, view, edit, and delete drives.

- Vaccination Status: Ability to update and view vaccination statuses.

- Reporting: Ability to generate reports with filters and export options.

## Testing

- Unit Tests: For individual functions and components.

- Integration Tests: For API endpoints.

- End-to-End Tests: For user flows (to be implemented).

## Deployment

- Frontend: Deployed on Netlify/Vercel.

- Backend: Deployed on Heroku/Render.

- Database: MongoDB Atlas.

## Future Enhancements

- Authentication: Implement user roles and permissions.

- Notifications: Send reminders for upcoming vaccination drives.

- Analytics: Dashboard for vaccination statistics.