

Cafeteria Menu Display

1. Introduction

Team ID:LTVIP2025TMID29043

Architha – Frontend Developer

Architha – Backend Developer

Architha – Database & Deployment

2. Project Overview

Purpose:

To provide a digital platform to display the daily cafeteria menu on screens or online, making it easier for students, staff, or customers to view available food options in real time.

Features:

Real-time menu updates

Separate menus for breakfast, lunch, and snacks

Admin login for cafeteria staff to manage menus

Responsive design for display on TVs and mobile

Integration with calendar for special days/festivals

3. Architecture

Frontend:

Developed using React.js, it displays menus in a user-friendly layout. Uses hooks for state management and TailwindCSS for design.

Backend:

Built using Node.js and Express.js, it handles menu CRUD operations and user authentication for admins.

Database:

Uses MongoDB Atlas to store menu data with fields like date, meal type, and items.

4. Setup Instructions

Prerequisites:

Node.js, npm, MongoDB Atlas account

Installation:

1. Clone the repository
2. Navigate to client and run npm install
3. Navigate to server and run npm install
4. Set up .env file with DB and PORT details
5. Run both servers using npm start

5. Folder Structure

Client:

```
/client
|
|--- public
|
|--- src
|
|   |--- components
|   |--- pages
|   |--- App.js
```

Server:

/server

```
|— controllers
|— models
|— routes
|— server.js
```

6. Running the Application

Frontend:

```
cd client
npm start
```

Backend:

```
cd server
npm start
```

7. API Documentation

GET /api/menu – Fetch all menus

POST /api/menu – Add new menu item

PUT /api/menu/:id – Update menu by ID

DELETE /api/menu/:id – Delete menu by ID

Example response:

```
{  
  "mealType": "Lunch",  
  "date": "2025-08-04",  
  "items": ["Rice", "Dal", "Curry", "Curd"]  
}
```

8. Authentication

Admins login using JWT tokens

Token stored in HTTP-only cookie

Middleware to protect admin routes

9. User Interface

Simple, clean UI for cafeteria display

Mobile-responsive layout

Admin dashboard for menu management

10. Testing

Tools Used:

Jest (unit tests) and Postman (API testing)

Strategies:

Unit testing for API routes

Manual testing for UI interactions

11. Screenshots or Demo

Demo Link (if available)

Screenshots:

Daily menu display

Admin login form

Menu editor UI

12. Known Issues

Delay in menu refresh on low-speed networks

No print support (planned in future)

No multilingual support yet

13. Future Enhancements

SMS or email alert for today's menu

QR code scan to view menu on mobile

Menu feedback system

Integration with digital signage systems