FullStackDevelopmentwithMERN Project Documentation format

# Introduction

* + **ProjectTitle:**[CafeteriaMenuDisplay]
  + **TeamMembers:-**Potnuri Chaitanya Sai Avinash

Nakka Sai Surya Durga Mouli Lokesh

Mummidi Naga Chandini

Manyam DeviSowmya

# ProjectOverview

* + **Purpose:**Cafeteriamenudisplayistoinformcustomersofavailablefoodoptions, prices, and specials clearly and efficiently.
  + **Features:**Dynamicmenuupdates,real-timepricing,nutritionalinfo,visualappeal, and user-friendly interface

# Architecture

* + **Frontend:**Leveragescomponent-baseddesigntorenderdynamicmenuitems, handle state management, and enable real-time updates viaAPIs.
  + **Backend:**HandlesAPIrouting,managesmenudatawithadatabase,andsupports real-time updates and admin controls.
  + **Database:**Collectionslikemenus,items,andstatus,whereinteractions include CRUD operations to fetch, update, and display daily menu items

dynamicallyonthefront-end.

# Setup Instructions

* + **Prerequisites:**Listsoftwaredependencies(e.g.,Node.js,MongoDB).
  + **Installation:**Step-by-stepguidetoclone,installdependencies,andsetupthe environment variables.

# FolderStructure

* + **Client:**MenuNAME,MenuItem,andstatus
  + **Server:**Expresswithmodularroutes,controllersforhandlinglogic,andaMongoDB database connection through Mongoose for managing cafeteria menu data.

# RunningtheApplication

Provide commands to start the frontend and backend servers locally.

* **Frontend:**npmstartintheclient directory.
* **Backend:**nodeserver.jsintheserverdirectory.

# APIDocumentation

* + GET/menu,POST/menu,PUT/menu/:id,DELETE/menu/:id,and optionally GET /menu/category/:category
  + **GET/menu**–fetchallitems,noparams,returns[{id,name,price,category

}];

**POST/menu**–createitem,body:{name,price,category},returns{id, name, price, category };

# Authentication

* + ArehandledusingJSONWebTokens(JWT),whereuserslogintoreceiveatoken that must be included in request headers to access or modify protected routes.
  + JWTtokensaregenerateduponlogin,storedontheclientside(e.g.,inlocalStorage),and

sentintheAuthorizationheader(Bearer<token>)witheachrequesttovalidateuser sessions and permissions.

# UserInterface

1. **Testing**

Strategyincludesunittestswith**Jest**forbackendlogic,**ReactTestingLibrary** for frontend component testing, and **Postman** or **Supertest**forAPI endpoint validation.

# ScreenshotsorDemo

<https://drive.google.com/file/d/1crZQWEcNlTP5C6mm4nU2V6qSIr8oBal7/view?usp=sharing>

# KnownIssues

IncludeoccasionalAPI delays,lackofinputvalidationonsomeforms, and

missingrole-basedaccesscontrolwhichmayallowunauthorizededitsiftokens are exposed.

# FutureEnhancements

Futureimprovementscouldincludeaddingrole-basedaccesscontrol,real-time menuupdateswithWebSockets,mobileappintegration,andenhancedUIwith filtering and search capabilities.