**PROJECT TITLE** : CookBook: Your Virtual Kitchen Assistant

1.Introduction

**PROJECT TITLE** : CookBook: Your Virtual Kitchen Assistant

**Team Members**:

1.S.ASMIYA

2.A.ARSHINA AFSAN

3.N.AFROSE ZAMEERA

4.T.BAVADHARINI

5.P.ANUSHKA

**2. Overview:**

**Purpose:**

This cookbook guides you through building a Node.js-based AI-powered platform that augments every phase of the Software Development Lifecycle (SDLC). The platform integrates AI tools into requirements analysis, code generation, test creation, bug detection, and performance monitoring.

**Goals:**

\*Accelerate development through automation

\*Reduce manual errors and guesswork

\*Offer AI-driven insights to all SDLC roles

**3.. System Architecture (Node.js Stack):**

**Frontend**

Tech: React + Material UI

Multi-tab interface (Requirements, Code, Tests, Monitoring, Assistant)

Uses React Router for navigation

Charts: Chart.js / Recharts

Syntax Highlighting: Prism.js

**Backend**

Tech: Node.js + Express.js

REST API endpoints

Uses async/await for non-blocking I/O

Swagger (via swagger-ui-express) for API docs

Multer for file uploads (e.g., .txt or .docx for requirements)

**AI Integration**

**Task Tool / Model**

Requirement Parsing Hugging Face Transformers (NER models via Python microservice)

Code Generation OpenAI Codex / Code LLMs via API

Test Generation Custom AI models or Codex

Bug Detection Python service using Isolation Forest / LSTM

DevOps Prediction Time-series models (ARIMA/Prophet via microservices)

Assistant OpenAI ChatGPT API (Node.js wrapper)

**4**. **Setup Instructions**

**Prerequisites**

Node.js v18+

npm or Yarn

Python (for ML microservices)

MongoDB / PostgreSQL (optional)

OpenAI / IBM Watsonx / Hugging Face API keys

Docker (optional, for containerization)

**Installation Steps**

git clone https://github.com/your-org/ai-sdlc-node.git

cd ai-sdlc-node

# Install dependencies

npm install

# Create a .env file

cp .env.example .env

# Add API keys and environment configs

# Start backend

npm run dev

# In another terminal, start frontend

cd client

npm install

npm start

**5. Folder Structure**

ai-sdlc-node/

├── backend/

│ ├── routes/

│ │ ├── requirements.js

│ │ ├── codegen.js

│ │ ├── tests.js

│ │ ├── bugs.js

│ │ └── monitor.js

│ ├── services/

│ │ └── aiModels.js

│ ├── controllers/

│ │ └── \*.js

│ ├── utils/

│ └── server.js

├── client/

│ ├── src/

│ │ ├── pages/

│ │ │ ├── Requirements.jsx

│ │ │ ├── CodeGen.jsx

│ │ │ ├── Tests.jsx

│ │ │ ├── Monitor.jsx

│ │ │ └── Assistant.jsx

│ └── App.jsx

├── .env

├── README.md

└── package.json

**6. Running the Application**

1**. Start the Express backend:**

npm run dev

**2. Start the React frontend:**

cd client

npm start

**3. Access via browser at:**

<http://localhost:3000>

**Navigate Tabs:**

Upload Requirements

Generate Code/Test

Detect Bugs/Anomalies

View Performance Matrics

Ask AI Assistant

**7. API Documentation (Swagger UI)**

**ENDPOINT METHOD DESCRIPTION**

/api/requirements/parse POST Extracts structured requirements

/api/codegen POST Returns Node.js code snippets

/api/tests POST Generates test cases (Jest, Mocha, etc.)

/api/bugs POST Analyzes logs/code for bugs

/api/monitor GET Predicts system metrics

/api/chat POST AI chatbot for SDLC queries

Swagger UI: http://localhost:8000/docs

**8. Authentication**

For demo purposes, routes are open. For production:

JWT-based user sessions

Role-based access control

API key verification

OAuth2 integration for SSO (Google, Github)

9**. User Interface Highlights**

Sidebar navigation for SDLC stages

Responsive tabbed layout

Code & Test viewers with syntax highlighting

Real-time charts and logs

Download/export buttons (JSON, CSV, TXT)

**10. Testing**

Unit Tests: Mocha/Chai for service logic

API Tests: Postman or Supertest

Mock Tests: Sample requirement documents

Error Handling: Edge cases & timeouts

CI/CD: GitHub Actions to run tests and linting

**11.Screenshots (To Add)**

Dashboard view

Code generation UI

AI Assistant that

Monitoring dashboard with charts

Test case viewer

**12. Known Issues**

AI code generation accuracy depends on prompt quality

Performance issues with large documents (>5k words)

Anomaly detection requires labeled log datasets

AI assistant may forget context mid-session

**13. Future Enhancements**

GitHub/GitLab integration for auto PRs

AI-generated CI/CD pipelines

Intelligent refactoring assistant

Auto-documentation tools

Multi-language code support

Model fine-tuning for domain-specific SDLCs

Let me know if you'd like:

A Markdown file (.md) version

A PDF version for submission

Help converting this into a deployment-ready app

Any code samples or boilerplate to go along with this guide.

**14.Coclusion:**

**With this cookbook program, you've embarked on a delicious adventure, exploring new recipes, techniques, and flavors. Whether you're a seasoned chef or a kitchen novice, we hope these recipes have inspired you to create memorable meals and moments with loved ones.**