

INSIGHT STREAM : NAVIGATE THE NEWS LANDSCAPE

Project submitted to the Bharathidhasan University, Tiruchirappalli

In partial fulfillment of the requirements for the award of the degree of

BACHELOR OF SCIENCE IN COMPUTER SCIENCE

Submitted by

Name : B.KEERTHIKA

Reg. No: 24057011802112038

NM ID : A02BE50C9102C5ABF814D1C4ADBD12AB

Under the Guidance of

Dr. M. RAJENDIRAN M.C.A., M.Phil., Ph. D



Department of Computer Science

Government Arts and Science College

Jayankondam-621801

November – 2025

Department of Computer Science

Government Arts and Science College

Jayankondam-621801



CERTIFICATE

This is to certify that the project entitled by INSIGHT STREAM : NAVIGATE THE NEWS LANDSCAPE is a bonafide report done by B.KEERTHIKA Reg No: 24057011802112038 in partial fulfillment of the requirements for the BACHELOR OF SCIENCE IN COMPUTER SCIENCE offered by the Department of Computer Science, Government Arts and Science College, Jayankondam during the academic year 2024 to 2027.

Faculty Mentor Head of the Department

The Viva-Voce held on …………… at the Department of Computer Science, Government Arts and Science College, Jayankondam.

Internal Examiner External Examiner

ACKNOWLEDGEMENT

I would like to thank Dr. M. RAJAMOORTHY, **M.Sc., M.Phil., Ph.D.,** Principal, Government Arts and Science College, Jayankondam for providing me an opportunity and environment that made the work possible.

I would like to express my deep sense of gratitude to Dr.G. KARTHIKEYAN, M.Sc., M.Phil., Ph.D., Head, Department of Computer Science, Government Arts and Science College, Jayankondam for giving me valuable support to complete this study successfully.

I would like to extend my heartfelt thanks to my guide Dr. M.RAJENDIRAN, M.C.A.,M.Phil., Ph.D Department of Computer Science, Government Arts and Science College, Jayankondam for an efficient guidance throughout the course of my project work.

I also express my thanks to my parents, my friends and well-wisher for the encouragement and best

wishes in the successful Completion of the project.

COOK BOOK : YOUR VIRTUAL KITCHEN ASSISTANT

## INTRODUCTION



1

Project Title : INSIGHT STREAM : NAVIGATE THE NEWS LANDSCAPE

Team ID : NM2025TMTD34974

Team Leader : B. KEERTHIKA&victoriyakeerthi2007@gmail.com

Team Members:

1. [T.DEEPA&deephandeephan18@gmail.com](mailto:T.DEEPA&deephandeephan18@gmail.com)

1. [E.EMAYADEVI&emayadevi13@gmail.com](mailto:E.EMAYADEVI&emayadevi13@gmail.com)
2. [S.KALKI&kalkisankar953@gmail.com](mailto:S.KALKI&kalkisankar953@gmail.com)

# PROJECT OVERVIEW



2

1. **PURPOSE:**

Cook Book is a virtual kitchen assistant designed to help users discover,

organize, and create recipes with ease. It provides personalized meal suggestions, step-by-step cooking guidance, and smart kitchen management to make cooking more enjoyable and efficient.

# Key Features :

* + **Recipe Discovery & Recommendations**
    - **Browse avast collection of recipes filtered by cuisine, dietary preferences, and difficulty level.**
    - **AI-powered personalized recommendations based news preferences and party items.**
  + **Smart Party & Ingredient Management**
    - **Track available ingredients and get recipes suggestions based on**
    - **what’s on hand.**
    - **Automated shopping list creation and inventory reminders.**
  + **Step-by-Step Cooking Assistant**
    - **Voice-guided on-screen instructions for hands-free cooking.**
    - **Built-in timers and measurement conversions.**
  + **Meal Planning &Nutrition Tracking**
    - **Weekly meal planner with calorie and nutrition breakdown.**
    - **Export meal plans or shopping lists for easy access.**
  + **Community& Sharing**
    - **Share your own recipes with others.**
    - **Rate, review, and save favorite recipes to a personal collection.**
  + **Admin Dashboard**
    - **Manage recipes, user content, and community feedback.**
    - **Analytics on popular dishes and user engagement.**

1. **ARCHITECTURE:**
   * **Frontend: Technology Stack:**

React.js (or Angular/Vue) for building anointer active and responsive interface. Tailwind CSS / Bootstrap for styling and layout.

* + **Backend: Technology Stack:**
* **Node.js with Express.js (or Python Flask/Django )handles server**
* **logic and APIs.**
* **Database: Mongo DB/My SQL to store user data, quiz results, and**

**news analysis content.**

* **Authentication: JWT (JSON Web Tokens) for secure login and**

**session handling.**

* + **Database: Database Choice:**
* **Mongo DB (No SQL)–for flexible, document-based storage of articles,**

User progress, and learning modules.

* **(Alternative: My SQL/Postgre SQL if relational structure is preferred).**

1. **SETUP INSTRUCTIONS:**
   * **Prerequisites:**
2. **Software Requirements :**

Node.js (for back end server and package management)

Express.js/Flask/Django (depending on chosen back end framework) Mongo DB / My SQL (for database management)

React.js/Angular/Vue.js (for front end development) Git hub (for version control and collaboration)

Visual Studio Code(VS Code) or any suitable IDE

1. **System Requirements**

Operating System: Windows/Linux/macOS

Minimum RAM: 4 GB (8 GB recommended)

Processor: Intel i3 / AMD equivalent (i5 or higher recommended)

Storage: Atleast: 500 MB free space for dependencies and database

1. **Additional Dependencies**

npm (Node Package Manager) for installing frontend & backend libraries Browser(Chrome/Firefox/Edge)for running and testing front end modules Postman / Thunder Client for testing API endpoints

* + **Installation Steps:**

1. **Clone the Repository**

**Git clone<repository\_url> cd insight stream**

1. **Frontend Setup cd client**

npm install #Install front end dependencies

npm start #Start the front end development server

1. **Backend Setup cd ../server**

npm install #Install back end dependencies npm start # Start the backend server

1. **Database Setup**

Install and start Mongo DB (or MySQL, if chosen).Create a database named insightstream db.Configure connection details inside server/config/db.js (for MongoDB) or .env file.

1. **Access the Application Open your browser and visit:**

[http://localhost:3000](http://localhost:3000/)Login/Registertoexplorefeatures.

1. **FOLDER STRUCTURE:**

**│**

├──client/ #Frontend (React.js or chosen framework)

│├── public/ #Static files(HTML, images, icons)

│├──src/ #Source code

││├── components/#Reusable UIcomponents (Navbar,Footer,Card

││├── pages/ #Mainpages (Dashboard, News Analysis, Quizzes)

││├── assets/ #Images, styles, fonts

││├── services/ #API calls to backend

││└── App.js #Root React component

│└── package.json #Front end dependencies

**│**

├──server/ #Backend(Node.js/Express.js or Python)

│├── routes/ #API route definitions (users, news, quizzes)

│├── models/ #Database models(User, Article, Quiz, Reports)

│├── controllers/ #Handles logic for each route

│├── config/ #Database connection & environment variables

│└── server.js #Entry point for backend

**│**

├──database/ #Database scripts and seed data

│└── schema.sql/json #Schema orinitial data(if needed)

**│**

├──docs/ #Project documentation(purpose, features, ERD)

**│**

├──.env #Environment variables (DB credentials, API keys)

├──README.md #Project overview & instructions

└──package.json #Root dependencies (if mono repo setup)

1. **RUNNING THE APPLICATION:**
   * **Frontend:**

cd client npm start

* + **Backend:**

Cd server npm start

* + **Access:** [**http://localhost:3000**](http://localhost:3000/)

1. **API DOCUMENTATION:**
   * **User:**

POST/api/user/register→Register a new user

POST/api/user/login→ Authenticate user & return JWT token GET /api/user/profile/:id → Fetch user profile & progress

PUT/api/user/update/:id→Update user details DELETE /api/user/:id → Delete a user account

* + **Project:**

POST/api/projects/create→ Create a new learning projector analysis task GET /api/projects/:id → Fetch details of a specific project

GET/api/projects→ Fetch all available projects/modules PUT /api/projects/:id → Update a project or module DELETE /api/projects/:id → Remove a project

* + **Chats:**

POST/api/chat/send→ Send a message to another user

GET/api/chat/:user Id→ Fetch chat history with a specific user

GET/api/chat/conversations/:user Id→ Fetch all conversations for a user DELETE /api/chat/:chat Id → Delete a specific chat message

1. **AUTHENTICATION:**

Method Used:

* **JWT (JSON Web Token)–for secure user authentication and session handling.**

Flow:

1. **User Registration/Login**

When a user registers or login, credentials are verified against the database.

On success, the server genest easy JWT token containing user ID and role.

1. **Token Storage**

The JWT token issent to the frontend and stored in local Storage/ session Storage.

1. **Protected Routes**

Every request to protected APIs (e.g.,/api/news/analyze,/api/projects/create) must include the JWT token in the request header:

Authorization: Bearer<token>

1. **Token Verification**

The backend verifies the token using a secret key.

If valid→ request proceeds.

If invalid/expired→request is denied with 401 Unauthorized.

Security Features:

* + **Encrypted passwords using by before storing in database.**
  + **Token expiration (e.g.,1hour)to reduce mis user risks.**
  + **Role-based access:**

User:

Can analyze news, take quizzes, chat.

Admin:

Can manage projects, modules, and monitor reports.

1. **USER INTERFACE:**

Landing Page

1. **Hero sections how casing Cook Book’s purpose (“Your Virtual Kitchen Assistant”).**
2. **Highlights of key features: recipe discovery, smart pantry, meal planning.**
3. **Search bar for recipes right on the home page.**
4. **Call-to-action buttons: *Sign Up*, *Explore Recipes*, *Get Started*.**
5. **Test imonials or featured recipes.**

User Dashboard (instead of Free lancer Dashboard)

1. **Personalized recipe recommendations based on preferences and pantry items.**
2. **Quick access to saved/favorite recipes.**
3. **Meal planning tools and recent cooking history.**
4. **Notifications for expiring ingredients or trending recipes.**

Admin Panel

1. **Manage recipes, User accounts, and reported content.**
2. **Approve or curate community-submitted recipes.**
3. **Analytics for popular recipes, dietary trends, and engagement.**
4. **Content moderation for comments and reviews.**

Recipe Details Page (instead of Project Details Page)

1. **High-quality recipe images and a short description.**
2. **Ingredient list with “Add to Shopping List” or “Check Pantry” options.**
3. **Step-by-step instructions (with timers or voice guidance).**
4. **Nutrition facts and related recipes.**
5. **Reviews, ratings, and “Save to Favorites” button.**
6. **TESTING:**

Testing Approach:

* **Manual testing during development mile stones.**
* **API testing with Postman/Thunder Client.**
* **Browser-based testing for frontend modules (Chrome Dev Tools).**

Types of Testing Performed:

1. **Unit Testing**

Tested individual functions (e.g.,login validation, quizs coring).Backend routes tested for expected JSON responses.

1. **Integration Testing**

Verified communication between frontend and backend.

Example: Submitting a news article in frontend correctly triggers backend API and shows analysis results.

1. **Functional Testing**

Checked all features against requirements: User Registration/Login

News Analysis Fact-Checking Quizzes & Reports Chat Messaging

1. **UI/UX Testing**

Verified responsiveness across devices (desktop, tablet, mobile) Ensured accessibility (readable fonts, color contrasts, alt-texts).

1. **Security Testing**

Password encryption verified with bcrypt.JWT authentication tested for protected routes. Invalid tokens rejected with 401 Unauthorized.

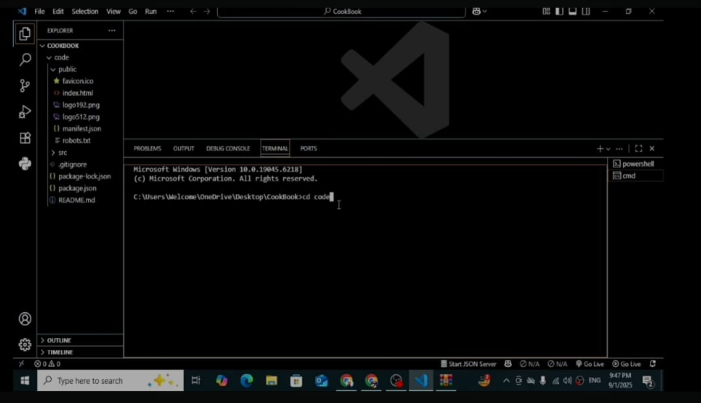
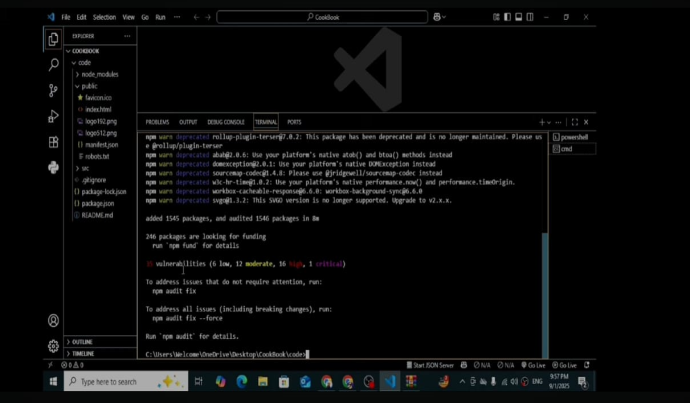
Testing Tools Used:

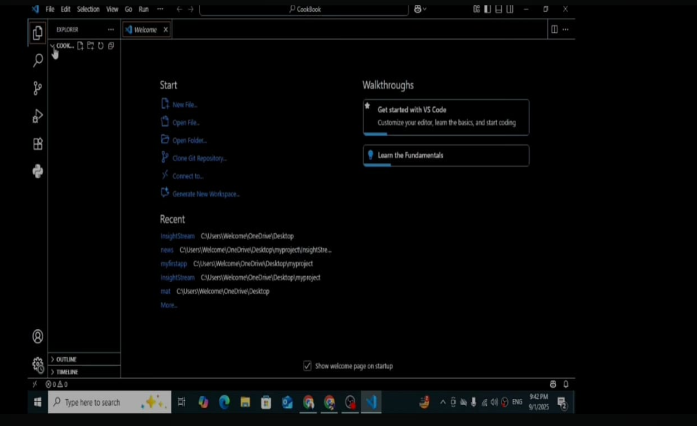
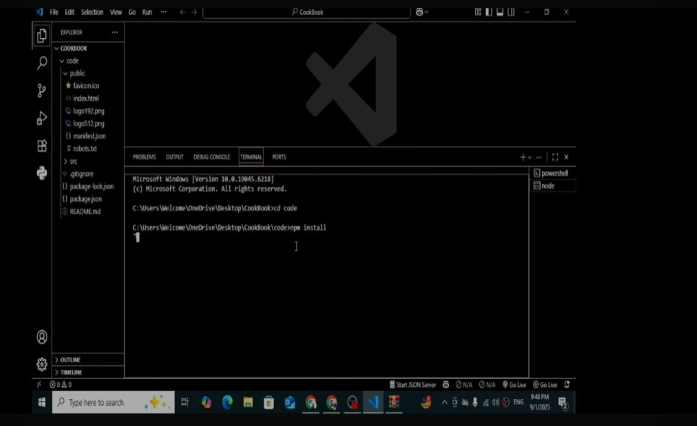
* + **Postman→ API request/response validation**
  + **Chrome Dev Tools→ UI and performance testing**
  + **Mongo DB Compass/My SQL Work bench→ Database validation**
  + **Jest/Mocha (optional) →Automated unit tests for backend**

Outcome:

* + **All core modules (User, News Analysis, Fact-Checking, Quiz, Reports, Chat) passed functionality checks.**
  + **Minor UI bugs identified and fixed(alignment, form validations).**
  + **System is stable for end-user testing and deployment.**

1. **SCREENSHOTS OR DEMO:**





# KNOWN ISSUES

🛠Functional Issues

* + **Recipe Search Accuracy–Search results may not always match user filters (e.g., dietary restrictions or ingredient exclusions).**
  + **Ingredient Recognition–Pantry tracking may misidentify or fail to detect certain ingredient names or quantities.**
  + **Voice Assistant Reliability–Voice commands may notwork consistently across different accents or noisy environments.**
  + **Offline Access–Limited functionality with out internet connection (recipes and pantry data may not sync).**

📱User Interface/User Experience Issues

* + **Mobile Responsiveness–Some pages(e.g., Recipe Details)may display poorly on smaller screens.**
  + **Load Times–High-resolution recipe images can slow down page loading on slower networks.**
  + **Navigation Complexity–First-time users may find the dashboard crowded or unintuitive.**

🔒Security & Data Privacy

* + **User Data Protection–Potential risks if recipe data or user profiles aren’t encrypted or secured.**
  + **Third-Party Integrations–Issues may arise if using APIs for nutrition in foor shopping lists without robust error handling.**

🧰Performance &Scaling

* + **High Traffic Handling–Limited scalability during peak times (e.g., holiday cooking seasons).**
  + **Caching Issues–Old data may persist if caching isn’t properly managed for updated recipes or pantry status.**

🧪Testing & Compatibility

* + **Browser Compatibility–Certain features (like timers or drag-and-drop meal planning) may not work on all browsers.**
  + **Device-Specific Bugs–Voice guidance or timers may be have in consistently on older devices.**

# FUTURE ENHANCEMENTS

🧠AI & Personalization

* + **Advanced AI Recipe Generator–Suggest new recipes based on available ingredients or user preferences.**
  + **Dietary Goal Tracking–Integrate with fitness apps to suggest meals aligned with health goals (e.g., keto, gluten-free).**
  + **Smart Substitutions–Automatically suggesting redients waps for allergies or unavailable items.**

📱User Experience Improvements

* + **Augmented Reality (AR) Cooking Mode–Overlay instructions or ingredient placement in real time using a phone camera.**
  + **Offline Recipe Access–Enabled own loading of favorite recipes for offline use.**
  + **Gamification –Achievements, badges, or challenges to encourage exploring new dishes.**

🌐Community & Social Features

* + **Live Cooking Sessions–Hostor join live virtual cooking classes.**
  + **Recipe Collaboration–Allow multiple users to co-create or edit a recipe together.**
  + **Enhanced Sharing–Share meal plans or grocery lists directly with friends or family.**

🛒Integration & Smart Kitchen

* + **Smart Appliance Integration–Connect with devices like smart ovens, fridges,or voice assistants (Alexa, Google Home).**
  + **E-commerce Links–One-click ordering of missing ingredients from partner grocery stores.**
  + **Barcode Scanning–Scan pantry items to quickly update inventory.**

📊Analytics & Insights

* + **Cooking Trends Dashboard–Show trending recipes and seasonal suggestions.**
  + **Waste Reduction Tracking–Analyze pantry usage to reduce food waste.**

🔒Security & Accessibility

* + **Multi-language Support–Expand to support multiple languages and regional cuisines.**
  + **Accessibility Features–Voice navigation, high-contrast modes, and screen reader compatibility.**
  + **Enhanced Data Privacy–End-to-end encryption for user data and pantry inventory.**