**OrderOnTheGo: Your On-Demand Food Ordering Solution**

**1. INTRODUCTION**

**1.1 Project Overview**

QuickBite is a full-stack food ordering web application built with React.js, Node.js, Express.js, and MongoDB. It allows users to register, browse food items, add to cart, and place secure orders. Restaurants can log in, manage their menus, and view order histories through a dedicated dashboard. Admins oversee platform activities, including managing users, products, and approving restaurants. The system uses MongoDB Atlas for cloud database storage and Mongoose for schema modeling.  
Role-based authentication ensures secure access for users, restaurants, and admins.  
The UI is responsive and user-friendly, built with reusable React components.  
QuickBite is designed for scalability and can be extended with features like payments and delivery tracking.

* 1. **Purpose**

The purpose of this application is to allow users to explore food items, place orders, and manage their profiles while enabling restaurants to manage menus and orders, and allowing admins to monitor and control overall operations.

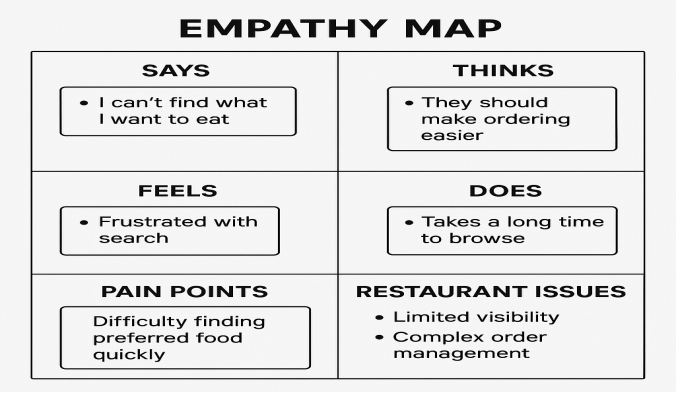
**2**. **IDEATION PHASE**

**2.1 Problem Statement**

Many users face difficulty finding their preferred food quickly on existing platforms.  
Cluttered interfaces and poor filtering lead to a frustrating ordering experience.  
Restaurants struggle with limited visibility and complex order management.  
There is a need for a smart, efficient, and user-friendly food ordering solution.

* 1. **Empathy Map Canvas**

Empathy maps were used to understand the pain points of users, such as difficulty finding preferred food quickly, and restaurant issues like limited visibility and complex order management.



SAYS -“I can’t find what I want to eat”

THINKS -“They should make ordering easier”

FEELS - Frustrated with search

DOES -Takes a long time to browse

PAIN POINTS - Difficulty finding preferred food quickly

RESTAURANT ISSUES -Limited visibilIty

* 1. **Brainstorming**

Ideas were brainstormed to provide a user-friendly interface, efficient search and filter options, and a role-based dashboard for all stakeholders.

**3. REQUIREMENT ANALYSIS**

**3.1 Customer Journey Map**

**Awareness Stage**

* User sees online ads, social media posts, or hears from a friend.
* Visits the app/website for the first time.

**Consideration Stage**

* Browses restaurant options and menu categories.
* Compares prices, ratings, delivery times, and offers.

**Decision Stage**

* Selects food items and adds them to the cart.
* Applies coupon codes or wallet discounts.

**Ordering Stage**

* Proceeds to checkout.
* Selects delivery address and preferred payment method.
* Confirms the order.

**Delivery Stage**

* Receives order tracking updates (real-time map, ETA).
* Delivery person arrives with the order.

**Post-Delivery Stage**

* User receives a request for feedback/review.
* May contact support in case of issues.

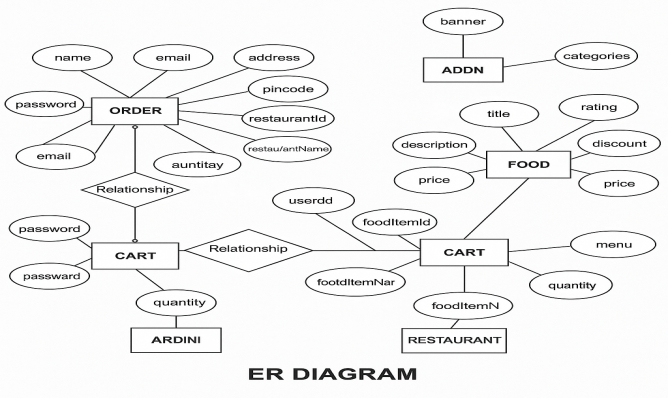
**3.2 Solution Requirement**

* Easy navigation, smart search, real-time order tracking, and secure payments.
* Tools for managing menus, orders, and delivery updates with real-time notifications.
* Manage users, restaurants, complaints, and promotional offers.
* Scalable backend, responsive frontend, real-time updates, and third-party integrations.

**3.3 Data Flow Diagram**

Data flows between users, restaurants, admin, and the MongoDB database with Node.js

serving as the mediator.



**3.4 Technology Stack**

Frontend: React.js  
Backend: Node.js + Express  
Database: MongoDB Atlas  
Authentication: JWT  
Version Control: Git

**4. PROJECT DESIGN**

**4.1 Problem Solution Fit**

Designed with users, restaurants, and admins in mind to address pain points like inefficient ordering and poor restaurant visibility.

**4.2 Proposed Solution**

The proposed solution is a responsive, full-stack web application designed to simplify online food ordering for users and streamline operations for restaurants. It allows users to browse popular dishes, add items to a cart, and place orders seamlessly. The application uses Next.js for the frontend and backend (API routes), and MongoDB Atlas as the database to store user data, dishes, and orders securely. The app is built with a modular and scalable architecture, ensuring ease of maintenance and future enhancements such as payment integration, order tracking, and real-time updates.

It also features:

* A visually appealing UI built with Tailwind CSS.
* Reusable components for better development efficiency.
* Mongoose models for smooth data communication with MongoDB.
* Secure use of environment variables for database connections.

**4.3 Solution Architecture**

* Presentation Layer (Frontend)
* Application Layer (Backend / API)
* Data Layer (Database – MongoDB Atlas)

**5. PROJECT PLANNING & SCHEDULING**

**5.1 Project Planning**

Sprint-1: 16 June 2025 - 18 June 2025: Requirement analysis and environment setup

Sprint-2: 19 June 2025 - 20 June 2025: Backend API development

Sprint-3: 21 June 2025 - 23 June 2025: Frontend UI and integration

Sprint-4: 24June 2025 - 26 June 2025: Testing and bug fixes

**6. FUNCTIONAL AND PERFORMANCE TESTING**

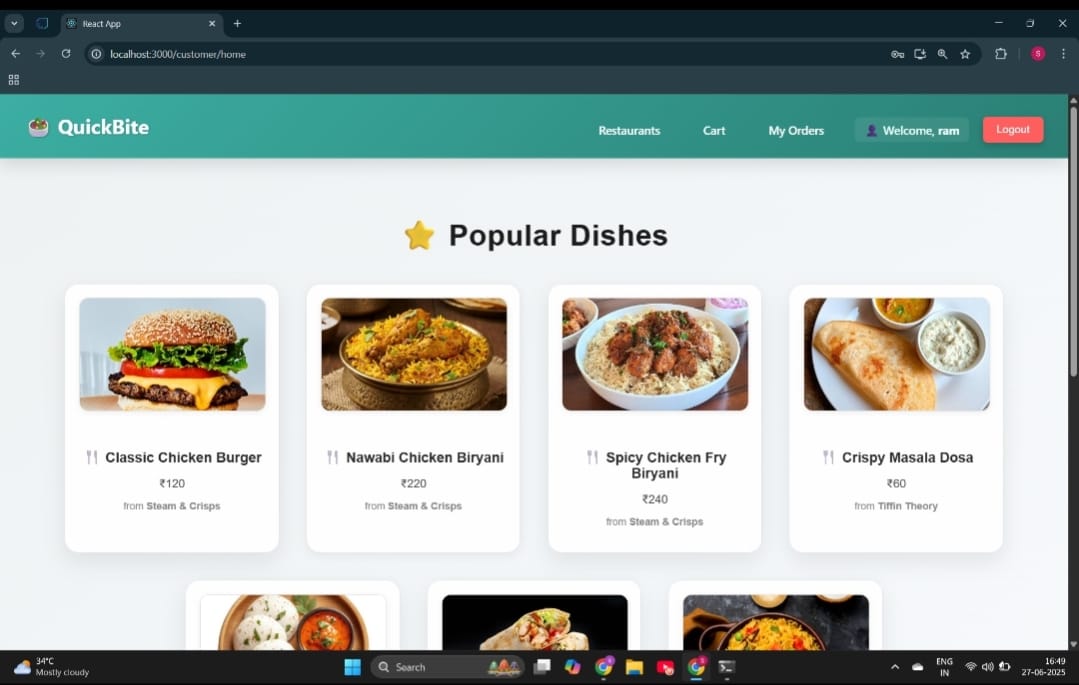
**6.1 Performance Testing**

* Tested for response time of APIs
* page load time
* MongoDB query performance.
* performance testing for this app

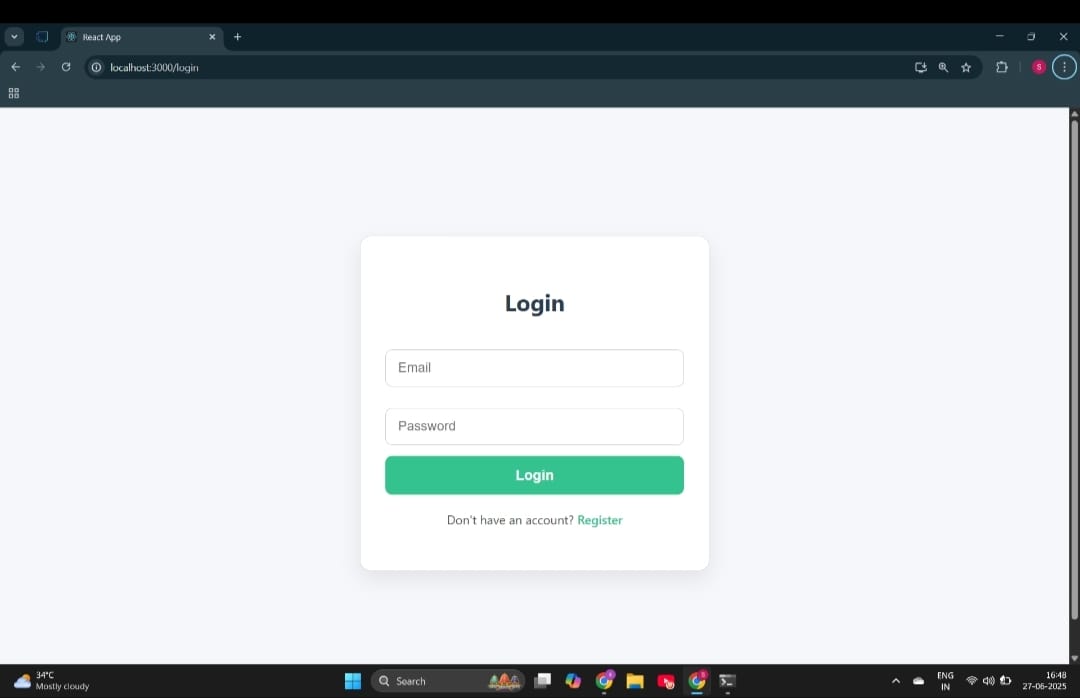
**7. RESULTS**

**7.1 Output Screenshots**

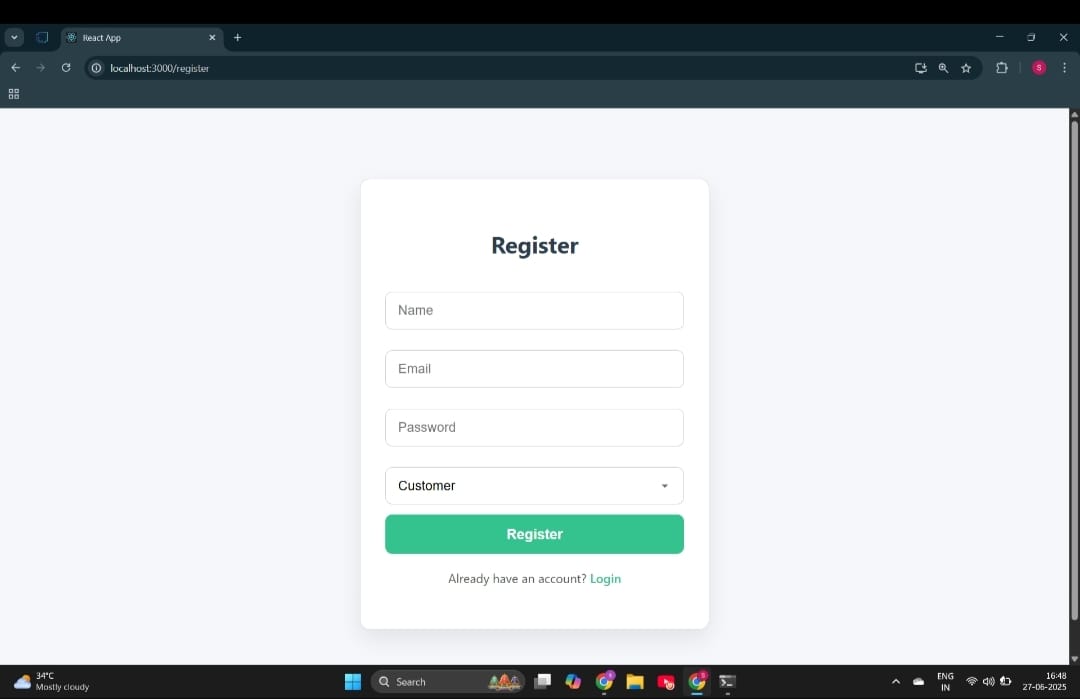
* Home page:



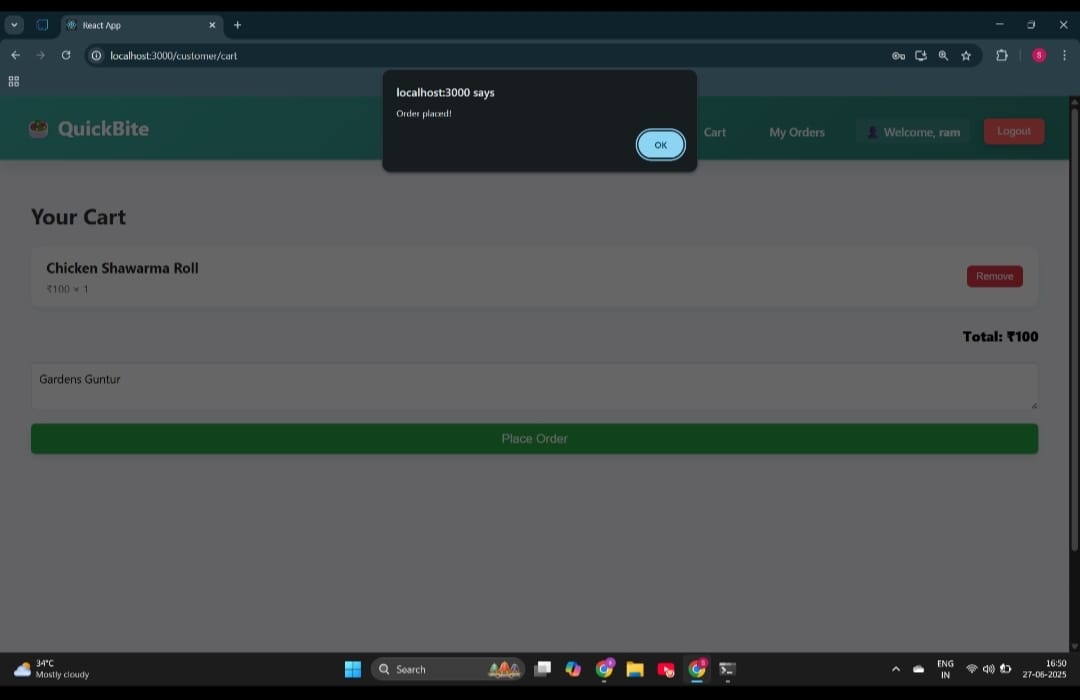
* Login Page:



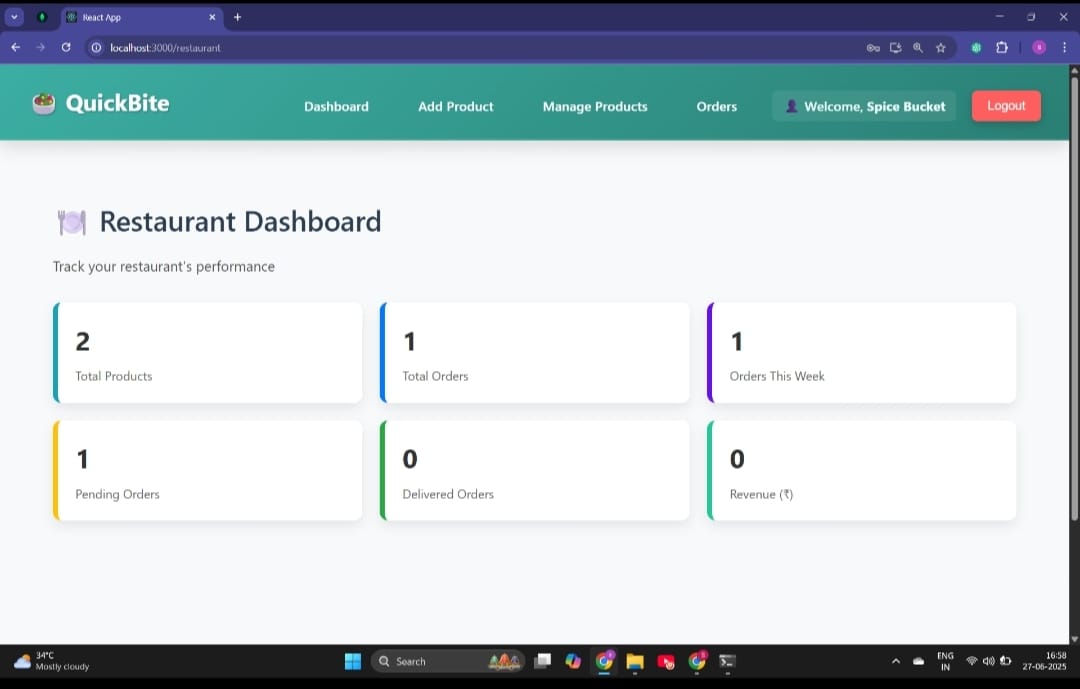
Register Page:



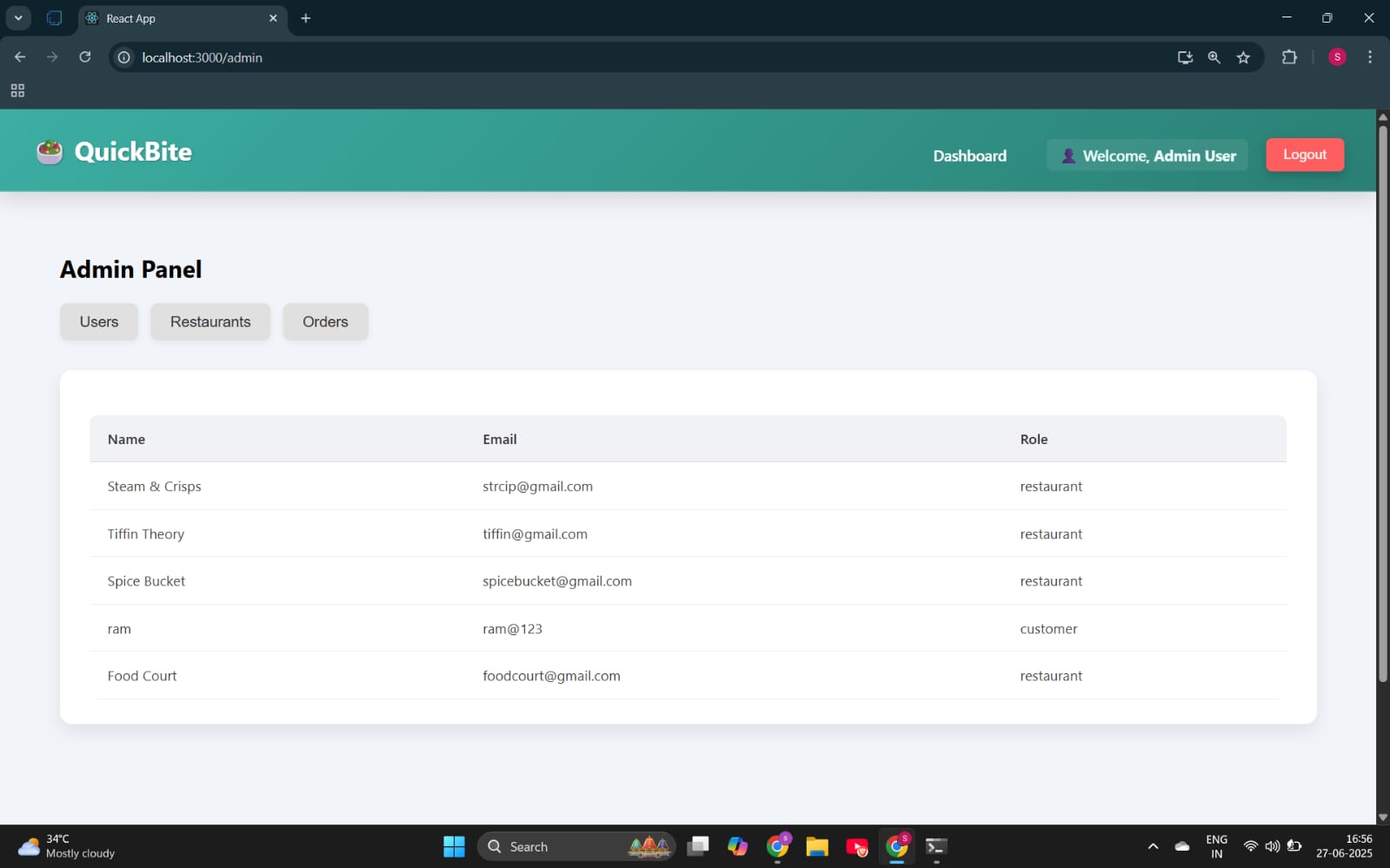
* Order placed:



* Restaurant Dashboard:



* Admin Panel:



**8. ADVANTAGES & DISADVANTAGES**

**Advantages:**  
- Real-time ordering  
- Admin and restaurant dashboards  
- Scalable backend  
  
**Disadvantages:**  
- Internet dependency  
- Initial database setup time

**9. CONCLUSION**

SB Foods is a modern and user-friendly food ordering web application designed to offer a seamless experience for users, from browsing their favorite dishes to placing orders and enjoying quick doorstep delivery. By integrating essential features like popular dish listings, real-time cart updates, secure checkout, and MongoDB-based data storage, the app ensures both convenience and reliability.

With a clean interface built using Next.js, and scalable backend integration via MongoDB Atlas, SB Foods not only addresses customer needs but also lays a strong foundation for future growth — including features like order tracking, feedback, and admin management. Overall, this project demonstrates how a full-stack web app can efficiently solve a real-world problem like food ordering, making the process simple, fast, and enjoyable for users.

1. **FUTURE SCOPE**

* Mobile app version
* payment gateway integration
* delivery tracking
* role-based analytics.

**11. APPENDIX**

Source Code:

Foodorderinghttps://drive.google.com/drive/folders/1vsQCzFTSorPLA8uwucw6TB4y3huXSgjV  
  
Demo Video: [Insert Demo Link Here]  
Dataset: Not applicable