**PROJECT REPORT**

1. **INTRODUCTION**

### Introducing CraveKart, the cutting-edge digital platform poised to revolutionize the way you order food online. With CraveKart, your food ordering experience will reach unparalleled levels of convenience and efficiency.

### Our user-friendly web app empowers foodies to effortlessly explore, discover, and order dishes tailored to their unique tastes. Whether you're a seasoned food enthusiast or an occasional diner, finding the perfect meals has never been more straightforward.

### Imagine having comprehensive details about each dish at your fingertips. From dish descriptions and customer reviews to pricing and available promotions, you'll have all the information you need to make well-informed choices. No more second-guessing or uncertainty – CraveKart ensures that every aspect of your online food ordering journey is crystal clear.

### The ordering process is a breeze. Just provide your name, delivery address, and preferred payment method, along with your desired dishes. Once you place your order, you'll receive an instant confirmation. No more waiting in long queues or dealing with complicated ordering processes – CraveKart streamlines it, making it quick and hassle-free

### **PROJECT OVERVIEW**

### The purpose of the OrderOnTheGo project is to build a user-friendly, full-stack online food ordering platform that connects customers with local restaurants. The goal is to streamline the process of discovering restaurants, placing food orders, and managing them efficiently. It aims to provide seamless experiences for customers, restaurant owners, and administrators, ensuring quick access to food, better order tracking, and efficient restaurant management.

### FEATURES:

1. **Comprehensive Product Catalog:** CraveKart boasts an extensive catalog of food items from various restaurants, offering a diverse range of items and options for shoppers. You can effortlessly explore and discover various products, complete with detailed descriptions, customer reviews, pricing, and available discounts, to find the perfect food for your hunger.

2. **Order Details Page**: Upon clicking the "Place Order" button, you will be directed to an order details page. Here, you can provide relevant information such as your shipping address, preferred payment method, and any specific product requirements.

3. **Secure and Efficient Checkout Process:** CraveKart guarantees a secure and efficient checkout process. Your personal information will be handled with the utmost security, and we strive to make the purchasing process as swift and trouble-free as possible.

4. **Order Confirmation and Details:** After successfully placing an order, you will receive a confirmation notification. Subsequently, you will be directed to an order details page, where you can review all pertinent information about your order, including shipping details, payment method, and any specific product requests you specified.

In addition to these user-centric features, CraveKart provides a robust restaurant dashboard, offering restaurants an array of functionalities to efficiently manage their products and sales. With the restaurant dashboard, restaurants can add and oversee multiple product listings, view order history, monitor customer activity, and access order details for all purchases.

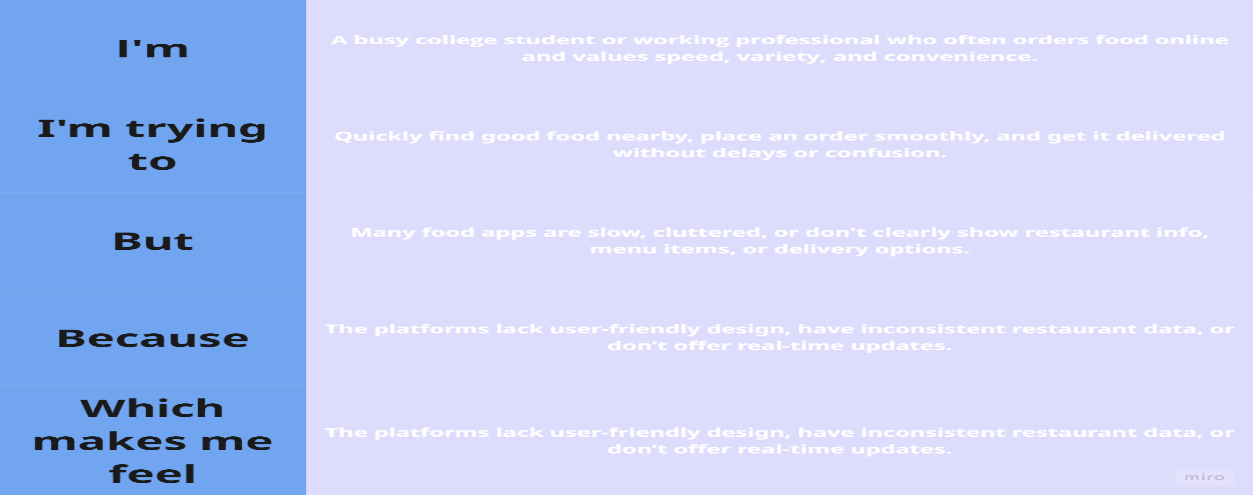
CraveKart is designed to elevate your online food ordering experience by providing seamless and user-friendly way to discover your desired foods. With our efficient checkout process, comprehensive product catalog, and robust restaurant dashboard, we ensure a convenient and enjoyable online shopping experience for both shoppers and restaurants alike.

1. **IDEATION PHASE**

**2.1 PROBLEM STATEMENT**

**Customer Problem Statement:**

Many customers today face frustration when ordering food online due to slow interfaces, confusing menus, lack of real-time updates, and limited restaurant options. They often struggle to find nearby restaurants that suit their taste, apply discounts easily, or complete an order without unnecessary steps. Customers expect a smooth, fast, and intuitive experience that allows them to explore food options, customize their orders, and receive accurate order updates. A platform that eliminates these pain points and makes the entire food ordering journey enjoyable and efficient is highly needed.



**Example:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Problem Statement (PS)** | **I am (Customer)** | **I’m trying to** | **But** | **Because** | **Which makes me feel** |
| PS-1 | A hungry student or working person looking for a quick meal | Find and order food from nearby restaurants easily | Many apps are confusing and slow, with too many steps | They lack intuitive design and real-time updates | Frustrated, impatient, and likely to abandon the order |
| PS-2 | A user who regularly orders food online | Explore restaurant options, check offers, and place an order smoothly | Offers aren’t visible clearly and restaurant info is outdated | The platform doesn't prioritize clarity and updated data | Disappointed and less likely to return to the platform |

**2.2 Empathy Map Canvas**

An empathy map is a simple, easy-to-digest visual that captures knowledge about a user’s behaviours and attitudes.

It is a useful tool to helps teams better understand their users.

Creating an effective solution requires understanding the true problem and the person who is experiencing it. The exercise of creating the map helps participants consider things from the user’s perspective along with his or her goals and challenges.

**Example:**

Diagram

Description automatically generated

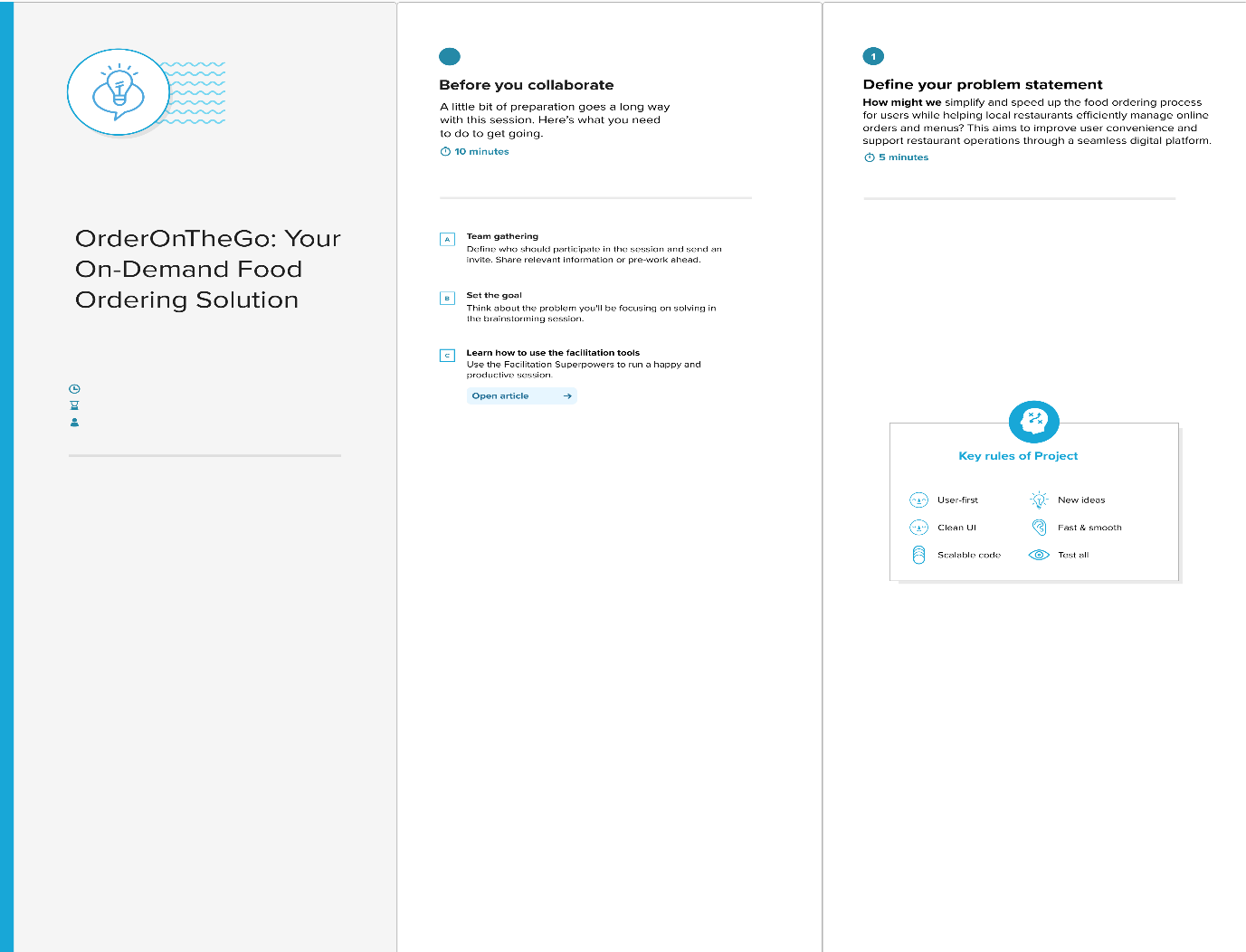
**Example: Food Ordering & Delivery Application**

Diagram

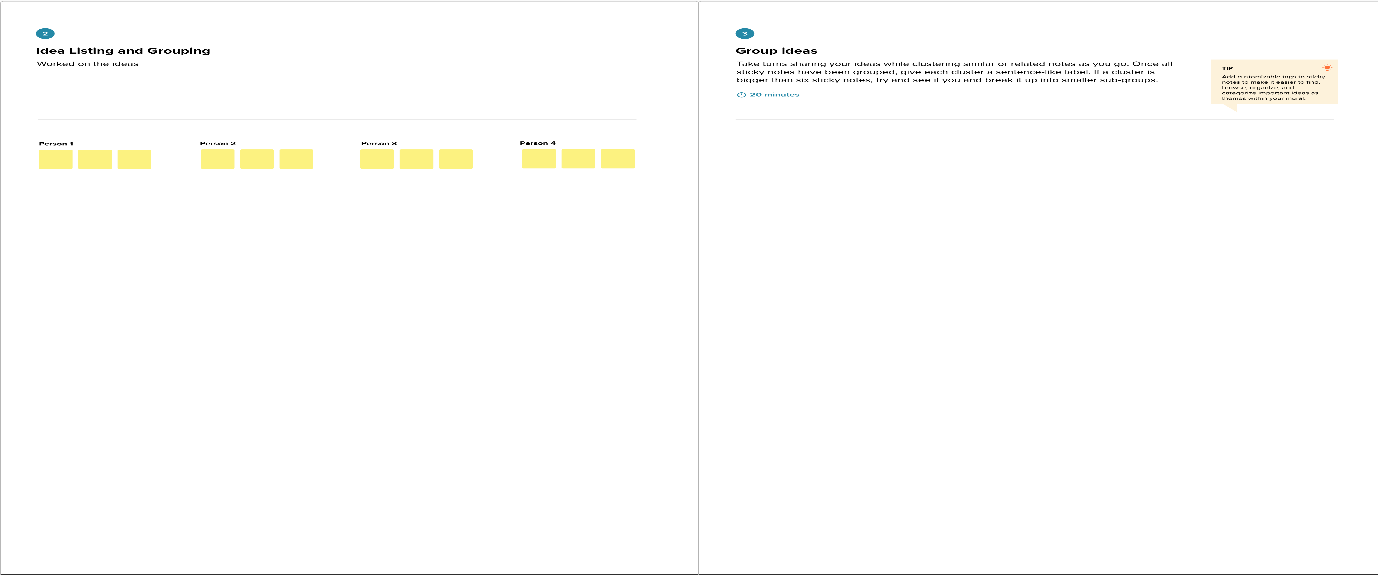
Description automatically generated

**2.3 Brainstorming**

**Step-1: Team Gathering, Collaboration and Select the Problem Statement**



**Step-2: Brainstorm, Idea Listing and Grouping**



**Step-3: Idea Prioritization**

A graph on a white sheet

AI-generated content may be incorrect.

3. **REQUIREMENT ANALYSIS**

**3.1 Customer Journey map**

**3.2 Solution Requirement**

**Functional Requirements:**

Following are the functional requirements of the proposed solution.

|  |  |  |
| --- | --- | --- |
| **FR No.** | **Functional Requirement (Epic)** | **Sub Requirement (Story / Sub-Task)** |
| FR-1 | User Registration | Registration through Form  Registration through Gmail  Registration through LinkedIN |
| FR-2 | User Confirmation | Confirmation via Email  Confirmation via OTP |
| FR-3 | Restaurant Browsing & Ordering | View all restaurants  View popular restaurants  View individual restaurant menu Add to cart Place order |
| FR-4 | Cart & Order Management | View/edit cart Remove item Place order View order history |
| FR-5 | Admin & Restaurant Dashboard | View/manage restaurants View/manage orders Approve restaurant signups |
| FR-6 | User Profile Management | View user profile Update profile info View past orders |

**Non-functional Requirements:**

Following are the non-functional requirements of the proposed solution.

|  |  |  |
| --- | --- | --- |
| **FR No.** | **Non-Functional Requirement** | **Description** |
| NFR-1 | **Usability** | The app should have a clean and intuitive UI (React + TailwindCSS), accessible by all user types. |
| NFR-2 | **Security** | User passwords are hashed (bcrypt), and JWT tokens are used for secure login and route protection. |
| NFR-3 | **Reliability** | MongoDB Atlas + Express.js ensures reliable data storage and consistent backend responses. |
| NFR-4 | **Performance** | Optimized API calls and efficient state updates ensure fast interactions, even with large menus. |
| NFR-5 | **Availability** | Hosted on cloud platforms to maintain 24/7 availability (can deploy to Render/Netlify). |
| NFR-6 | **Scalability** | Easily scalable with additional restaurant or user data due to use of NoSQL (MongoDB). |

**3.3 Data Flow Diagram**

**Diagram, timeline

Description automatically generated**

**3.4 Technology Stack**

**Technical Architecture:**

The Deliverable shall include the architectural diagram as below and the information as per the table1 & table 2

**Example: Order processing during pandemics for offline mode**

**Reference:** [**https://developer.ibm.com/patterns/ai-powered-backend-system-for-order-processing-during-pandemics/**](https://developer.ibm.com/patterns/ai-powered-backend-system-for-order-processing-during-pandemics/)



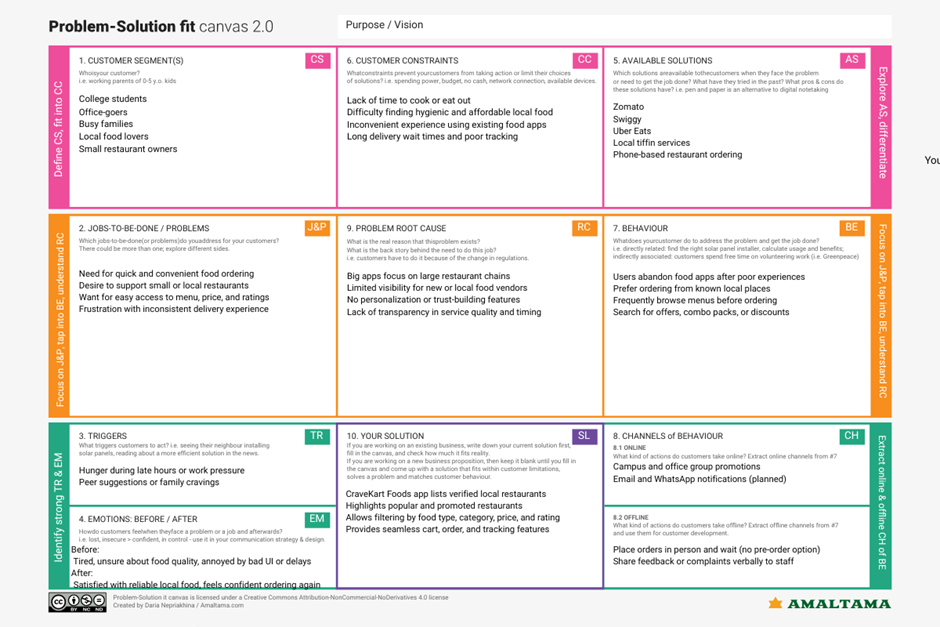
|  |  |  |  |
| --- | --- | --- | --- |
| **S.No** | **Component** | **Description** | **Technology** |
|  | User Interface | Web UI for user interaction | HTML, CSS, React.js, Tailwind CSS. |
|  | Application Logic-1 | Customer-side logic (e.g., menu browsing, cart, orders) | JavaScript (React.js), Axios |
|  | Application Logic-2 | Backend logic for authentication and food operations | Node.js, Express.js |
|  | Application Logic-3 | Admin & Restaurant logic (approvals, menu control, etc.) | Node.js, Express.js |
|  | Database | Stores all app data | MongoDB (Mongoose ODM) |
|  | Cloud Database | Hosted on cloud | |  | | --- | | MongoDB | |
|  | File Storage | Image hosting (restaurant images, product images) | Cloudinary (or local for dev) |
|  | External API-1 | Weather or geo-location API (optional for extensions) | Not used currently / OpenWeather API |
|  | External API-2 | Payment gateway (for future use) | *Razorpay / Stripe* (future integration) |
|  | Machine Learning Model | Not implemented in CraveKart Foods | N/A |
|  | Infrastructure (Server / Cloud) | Server deployment | **Render / Railway (for hosting)** or **Vercel (frontend)** |

**Table-2: Application Characteristics:**

|  |  |  |  |
| --- | --- | --- | --- |
| **S.No** | **Characteristics** | **Description** | **Technology** |
|  | Open-Source Frameworks | Frameworks used in client & server | React.js, Express.js, MongoDB, Node.js |
|  | Security Implementations | JWT auth, bcrypt hashing, CORS, rate limits. | bcrypt, JWT, CORS |
|  | Scalable Architecture | Separation of client/server, RESTful APIs, MongoDB Atlas for scaling | MERN Stack with REST APIs |
|  | Availability | Deployed on cloud platforms with high uptime (Vercel/Render/Railway) | Cloud hosting + MongoDB Atlas |
|  | Performance | React-based UI with lazy loading, API pagination, CDN usage (Vercel/CDN) | React, Axios, MongoDB Indexes |

**4. PROJECT DESIGN**

4**.1 Problem Solution Fit**



**4.2 Proposed Solution**

|  |  |  |
| --- | --- | --- |
| **S.No.** | **Parameter** | **Description** |
|  | Problem Statement (Problem to be solved) | Difficulty in discovering nearby restaurants, managing food orders online, and lack of a unified platform for users to browse, order, and track food delivery efficiently. |
|  | Idea / Solution description | CraveKart Foods is a web-based food ordering platform that connects customers with local restaurants. It allows users to register, log in, browse restaurants and menus, add items to cart, place orders, and track them—all through a user-friendly interface. |
|  | Novelty / Uniqueness | CraveKart Foods focuses on simplicity and real-time functionality. Unlike many apps, it uses optimized APIs for restaurant listings, live cart updates, and smooth UI flow. |
|  | Social Impact / Customer Satisfaction | It helps local restaurants reach more customers online and provides users with a convenient, time-saving food ordering experience. It also promotes hygienic food access through verified listings. |
|  | Business Model (Revenue Model) | Revenue can be generated via commission from restaurant partners, delivery charges, subscription plans for premium users, and advertisement slots on the platform. |
|  | Scalability of the Solution | The solution is built using modular and scalable backend architecture (Node.js + MongoDB) and a flexible frontend (React). It can be expanded to support mobile apps, multiple cities, and vendor dashboards in future. |

**4.3 Solution Architecture**

Solution architecture is a complex process – with many sub-processes – that bridges the gap between business problems and technology solutions. Its goals are to:

1. Find the best tech solution to solve existing business problems.
2. Describe the structure, characteristics, behavior, and other aspects of the software to project stakeholders.
3. Define features, development phases, and solution requirements.

Provide specifications according to which the solution is defined, managed, and delivered.

**5. PROJECT PLANNING & SCHEDULING**

**5.1 Project Planning**

**Product Backlog, Sprint Schedule, and Estimation (4 Marks)**

Use the below template to create product backlog and sprint schedule

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Sprint** | **Functional Requirement (Epic)** | **User Story Number** | **User Story / Task** | **Story Points** | **Priority** | **Team Members** |
| Sprint-1 | Registration | USN-1 | As a user, I can register using email and password | 2 | High |  |
| Sprint-1 |  | USN-2 | As a user, I receive a confirmation email after registration | 1 | High |  |
| Sprint-1 |  | USN-4 | As a user, I can register using Gmail | 2 | Medium |  |
| Sprint-1 | Login | USN-5 | As a user, I can log in with email and password | 1 | High |  |
| Sprint-2 | Browse Restaurants | USN-6 | As a user, I can browse restaurants by category | 2 | High |  |
| Sprint-2 | View Menu Items | USN-7 | As a user, I can view food items with ratings and prices | 2 | High |  |
| Sprint-2 | Cart | USN-8 | As a user, I can add/remove food items to/from cart | 3 | High |  |
| Sprint-2 | Order Placement | USN-9 | As a user, I can place an order and download invoice | 3 | High |  |
| Sprint-3 | Restaurant Dashboard | USN-10 | As a restaurant, I can add/edit/delete menu items | 3 | High |  |
| Sprint-3 | Restaurant Order Management | USN-11 | As a restaurant, I can view incoming orders | 3 | Medium |  |
| Sprint-3 | Admin Dashboard | USN-12 | As an admin, I can monitor users and restaurants | 2 | Medium |  |
| Sprint-3 | Admin Actions | USN-13 | As an admin, I can remove a user or restaurant | 2 | Low |  |
| Sprint-4 | Styling & Animations | USN-14 | As a user, I see smooth UI with animations | 2 | Medium |  |
| Sprint-4 | Testing & Bug Fixes | USN-15 | As a tester, I can test and log bugs | 3 | High |  |
| Sprint-4 | Deployment | USN-16 | As a developer, I can deploy the project online | 3 | High |  |
| Sprint-4 | Documentation | USN-17 | As a developer, I can prepare ReadMe and docs | 2 | Medium |  |

**Project Tracker, Velocity & Burndown Chart: (4 Marks)**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Sprint** | **Total Story Points** | **Duration** | **Sprint Start Date** | **Sprint End Date (Planned)** | **Story Points Completed (as on Planned End Date)** | **Sprint Release Date (Actual)** |
| Sprint-1 | 6 | 5 Days | 16 Jun 2025 | 20 Jun 2025 | 6 |  |
| Sprint-2 | 10 | 5 Days | 18 Jun 2025 | 22 Jun 2025 | 10 |  |
| Sprint-3 | 10 | 5 Days | 20 Jun 2025 | 24 Jun 2025 | 10 |  |
| Sprint-4 | 10 | 5 Days | 24 Jun 2025 | 29 Jun 2025 | 10 |  |

**Velocity:**

Imagine we have a 10-day sprint duration, and the velocity of the team is 20 (points per sprint). Let’s calculate the team’s average velocity (AV) per iteration unit (story points per day)

AV = sprint duration / velocity

= 36 / 4

=9

**6. FUNCTIONAL AND PERFORMANCE TESTING**

6**.1 Performance Testing**

**Project Overview:**

Project Name: OrderOnTheGo: Your On-Demand Food Ordering Solution

Project Description: A full-stack web application enabling users to discover restaurants, browse menus, place food orders, and download bills. The platform supports three user roles: Customer, Restaurant, and Admin.

Project Version: 1.0.0

Testing Period: 22-06-2025 to 27-06-2025

**Testing Scope:**

**Features to be Tested:**

* User registration/login (Customer, Restaurant, Admin)
* Browse restaurants by category
* View food items with prices, ratings, and discounts
* Cart management and order placement
* Invoice generation and download
* Restaurant panel for managing food items and orders
* Admin panel for monitoring users and restaurants

**User Stories/Requirements:**

* As a customer, I want to browse food items and place an order.
* As a restaurant, I want to manage food menus and view orders.
* As an admin, I want to view all users and restaurant activity.

**Testing Environment:**

* **URL/Location:** [http://localhost:3000](http://localhost:3000/)

**Test Cases:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test Case ID** | **Test Scenario** | **Test Steps** | **Expected Result** | **Actual Result** | **Pass/Fail** |
| TC-001 | Customer Login | Step 1: Navigate to login page  Step 2: Enter valid credentials  Step 3: Click login | Redirect to customer dashboard | Works as expected | Pass |
| |  | | --- | | TC-002 | |  | | Browse Restaurants | Step 1: Login as customer  Step 2: Click on a category  Step 3: View restaurants | Display restaurant list by category | Works as expected | Pass |
| TC-003 | Add to Cart and Order | Step 1: View food items  Step 2: Add items to cart  Step 3: Place order | Order placed and invoice generated | Order confirmed and bill downloaded | Pass |
| TC-004 | Restaurant Adds Food Item | Step 1: Login as restaurant  Step 2: Go to manage food  Step 3: Add new item | Food item added to restaurant menu | Food item visible in customer panel | Pass |
| TC-005 | Admin View Users | Step 1: Login as admin  Step 2: Click on users tab | Display user list | User list shown correctly | Pass |

**Bug Tracking:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Bug ID** | **Bug Description** | **Steps to reproduce** | **Severity** | **Status** | **Additional feedback** |
| BG-001 | Food image was not loading in menu | Step 1: Go to restaurant menu  Step 2: Check food item image | Medium | closed | Issue resolved – image uploads and displays correctly now |
| BG-002 | Invoice file name is unreadable | Step 1: Place order  Step 2: Download invoice | Low | Open | Needs clear filename format e.g. "Order\_1234.pdf" |

**Sign-off:**

Tester Name: K Abhinaya

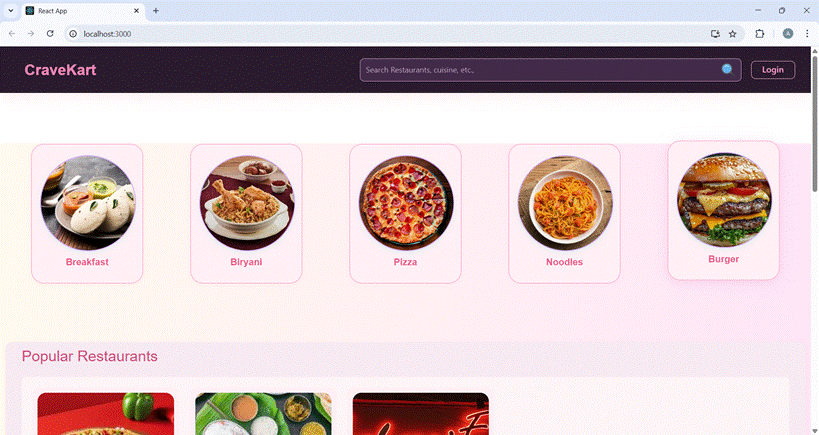
Date: 27-06-2025

Signature: Abhinaya

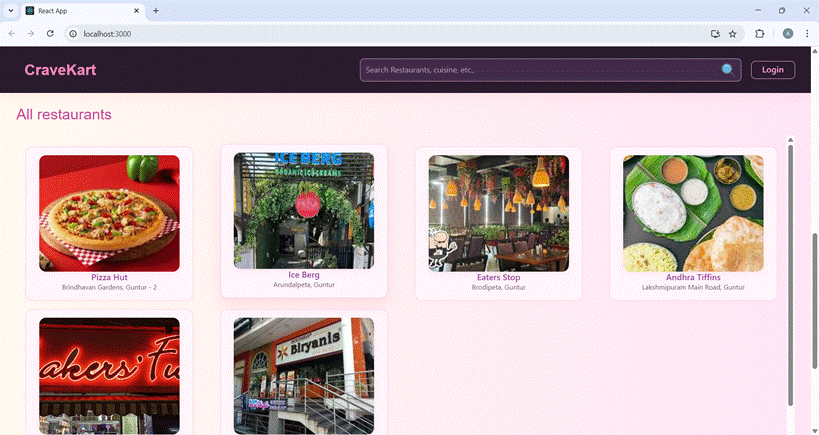
**7. RESULTS**

**7.1 Output Screenshots**

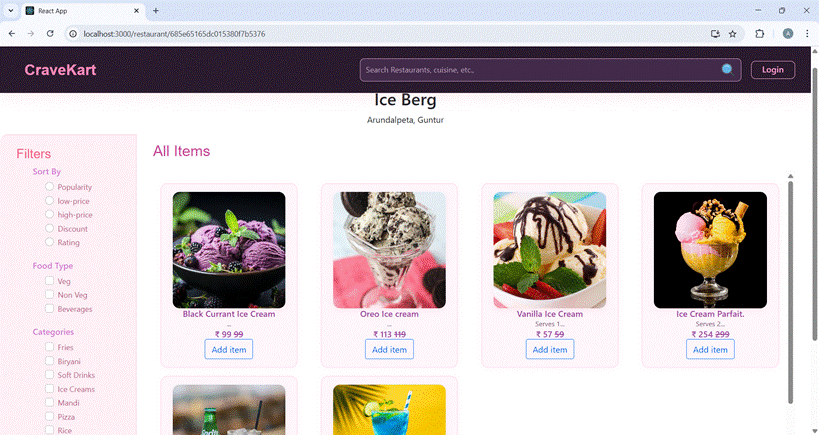
· **Landing page**



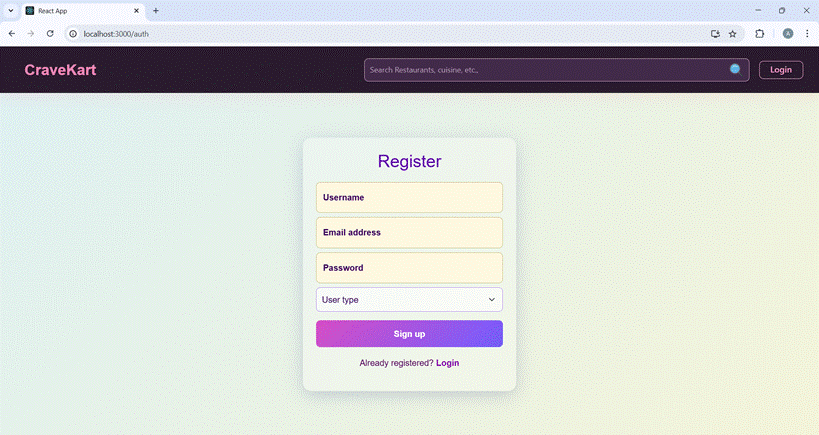
· **Restaurants**



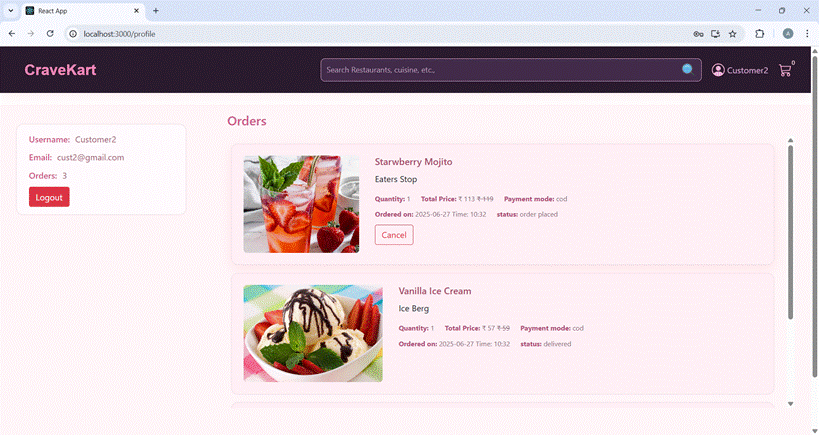
· **Restaurant Menu**



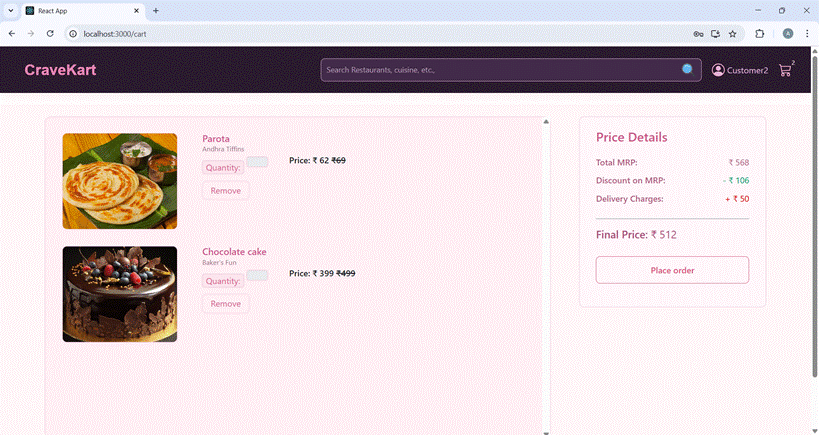
· **Authentication**



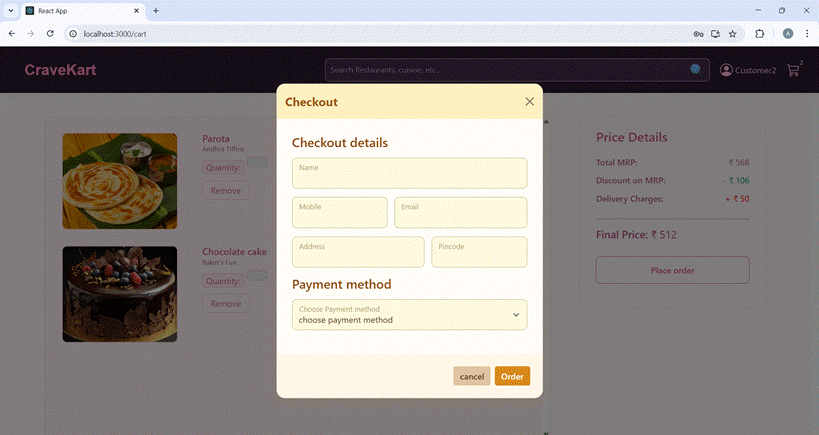
· **User Profile**



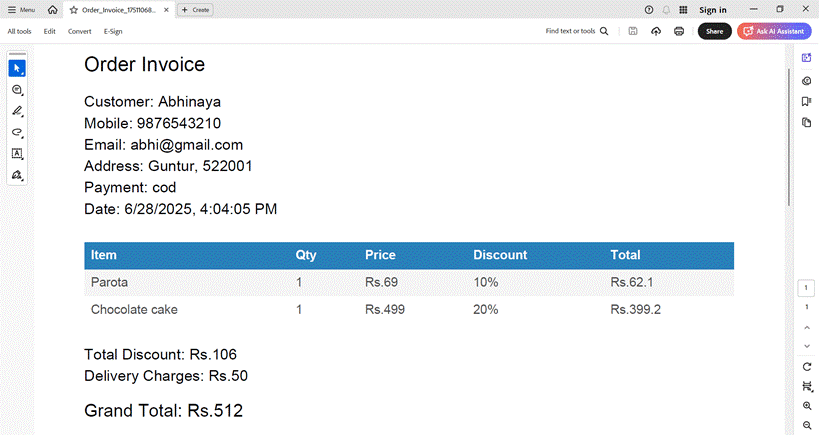
**· Cart**



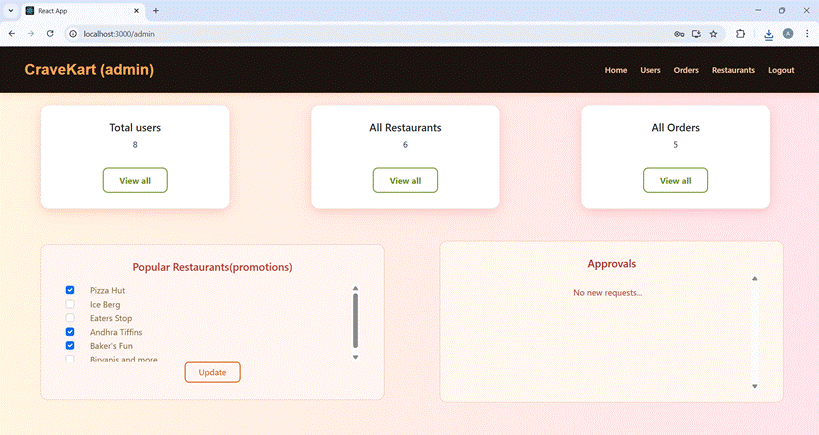
**· Checkout details**



**· Invoice Download**



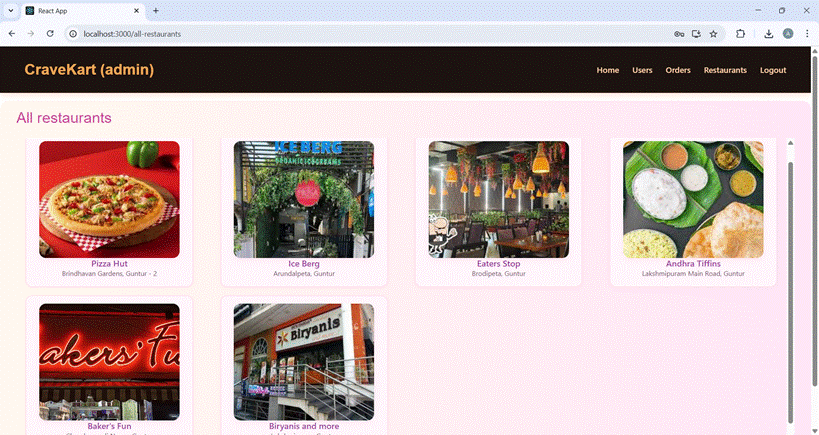
**· Admin dashboard**



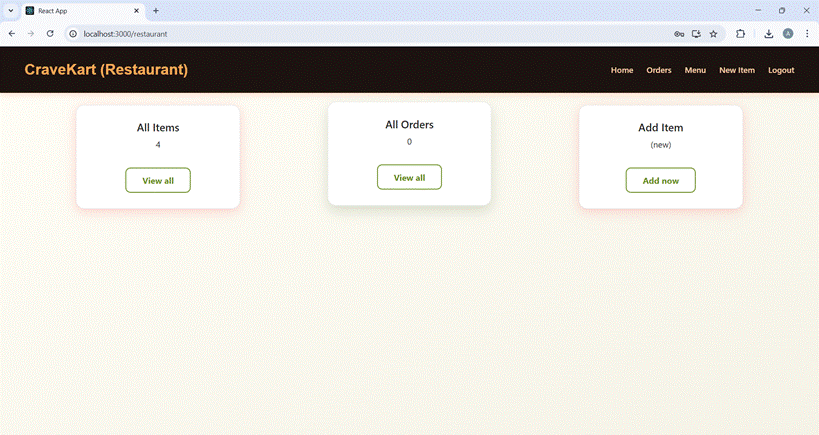
**· All Orders**



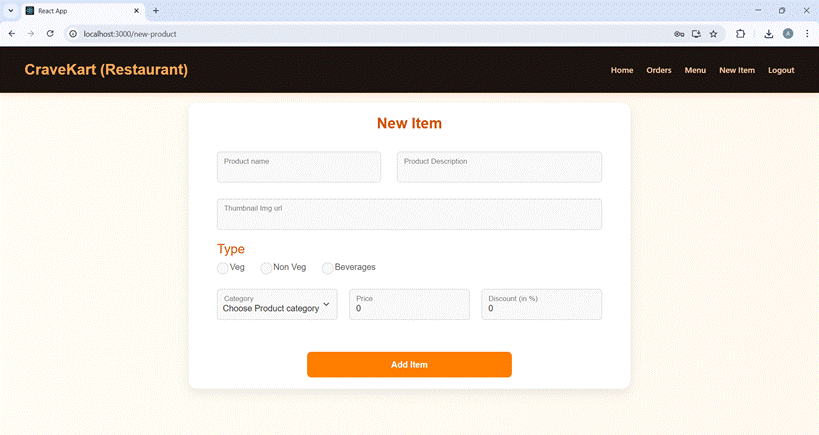
**· All restaurants**



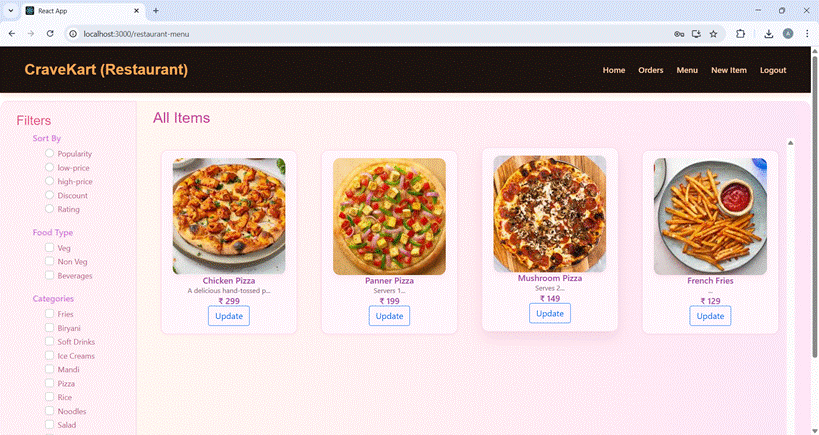
**· Restaurant Dashboard**



**· New Item**



All Items



**8. ADVANTAGES & DISADVANTAGES**

**Adavantages**

|  |  |
| --- | --- |
| **Advantage** | **Description** |
| **1. Simple & Cost-Effective** | Manual and black box testing require minimal setup and are ideal for small-to-mid projects. |
| **2. Real-User Validation (UAT)** | UAT ensures the system meets user needs in realistic scenarios. |
| **3. Flexibility** | Manual testing allows for spontaneous testing of new features and edge cases. |
| **4. Easy to Get Started** | Tools like Postman and browser DevTools are easy to use and require no advanced setup. |
| **5. Effective Role-Based Testing** | Each user type (Customer, Restaurant, Admin) was tested distinctly, ensuring role separation worked well. |

**Disadvantages**

|  |  |
| --- | --- |
| **Disadvantage** | **Description** |
| **1. Time-Consuming** | Manual testing takes longer and becomes inefficient as the application scales. |
| **2. Not Repeatable** | Manual test cases are harder to automate, track, or reuse in future test cycles. |
| **3. Prone to Human Error** | Manual checks can miss bugs that automation might catch. |
| **4. No Performance Testing** | Tools like Postman don’t measure speed or server load, missing out on performance issues. |
| **5. Limited Coverage** | Without automation, it's hard to test every combination of inputs, roles, and workflows consistently. |

**9. CONCLUSION**

The **OrderOnTheGo** project successfully delivers a responsive, user-friendly web application that simplifies online food ordering for customers while offering powerful tools for restaurants and administrators. The platform allows users to explore food options, manage their orders, and interact with restaurants seamlessly.

Throughout development, a structured Agile methodology was followed, with four well-planned sprints, detailed user stories, and thorough user acceptance testing. Features such as cart management, invoice generation, and role-based dashboards were implemented and tested successfully.

Manual and functional testing ensured the reliability of each module, and known issues were documented transparently. The project is now stable and ready for real-world use, with room for future enhancements like payment integration, real-time tracking, and mobile app development.

In conclusion, **OrderOnTheGo** stands as a complete, scalable food ordering solution that meets its core goals of convenience, speed, and usability for all users.

**10. FUTURE SCOPE**

### **1. Online Payment Integration**

* Add support for UPI, cards, wallets using Razorpay, Stripe, or Paytm.
* Enable real-time payment confirmation and secure transaction handling.

### **2. Real-Time Order Updates**

* Use WebSockets or Firebase to push live order status updates to users and restaurants.
* Reduce manual refresh dependency.

### **3. Mobile Application**

* Develop native or hybrid Android/iOS app using React Native or Flutter.
* Enable push notifications and geolocation-based restaurant discovery.

### **4. AI-Based Recommendations**

* Suggest restaurants or dishes based on user preferences and order history.
* Improve personalization using machine learning.

### **5. Multi-Language and Regional Support**

* Localize the platform for different languages and regions.
* Improve accessibility and widen the user base.

### **6. Analytics Dashboard**

* Provide data insights to admins and restaurants (orders, revenue, user trends).

**11. APPENDIX**

For any further doubts or help, please consider the GitHub repo,

<https://github.com/KalisettyAbhi234/OrderOnTheGo-Your-On-Demand-Food-Ordering-Solution/tree/main>

The demo of the app is available at:

<https://drive.google.com/file/d/1aT5kXVJX83ht5Yap2FwRCRcOqnS3_Mxb/view?usp=sharing>