



DineWise



Team 2 Pace Super Giants



TABLE OF CONTENTS



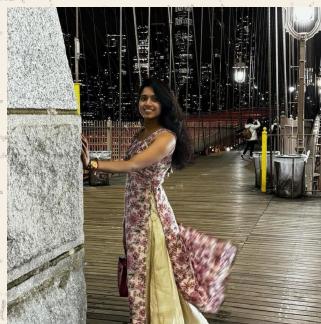
- **Team Member Roles and Responsibilities**
- **Improvements Made from Professor Feedback**
- **Personas**
- **Minimum Viable Product (MVP)**
- **Technologies**
- **Algorithms**
- **Diagrams**
 - Architecture Diagram
 - Context Diagram
 - ER Diagram
 - Sequence Diagram
 - State Diagram
 - Class Diagram
- **Product Backlog**
- **Sprint 1 Backlog**
 - Stories or Tasks Committed
 - Test Cases
 - Stories Completed
 - Stories Not Completed
- **Metrics**
 - Team Velocity - This Sprint
 - Average Velocity
 - Completed/Committed Ratio
 - Burndown Charts
- **Retrospective**
- **Sprint 2**
 - Stories Planned and Committed for Sprint 2
- **Project Demo - Sprint 1**
 - App Screenshot
 - API Slides
- **GitHub Link**
- **Live Application Demo**

OUR TEAM



★ **Deekshitha
Navuluri** ★

Full Stack Developer



★ **Panguluri
Bhavya** ★

Product Designer /
Frontend Developer



★ **Phanindhr
Thota** ★

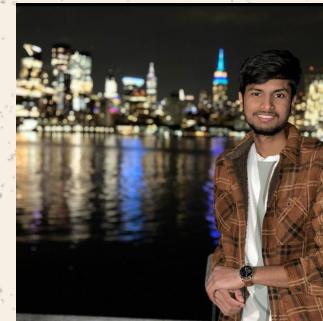
Frontend Developer

OUR TEAM



★ **Manoj Kumar
Ambavarapu** ★

Scrum Master /
Backend Developer



★ **Sai Mahesh
Sandeboina** ★

Machine learning
Engineer

Improvements

- Adding of story ids.
- Splitting up the stories which have 2 functionalities.
- Adding of user benefits.

Project Description

Project Name:	DineWise
Team:	Pace Super Giants
Project Description:	<p>For restaurant customers who wants personalized menu and even dynamic pricing options, the solution is the website which is machine learning based recommendation and handling dynamic pricing and this is an AI powered website that provides dishes based on your preference and pricing based on the demand unlike regular classic restaurants where they have single menu and even static pricing our application will provide personalized dine in experience with dynamic pricing.</p>
Benefit Outcomes:	<ul style="list-style-type: none">• This will improve the customer satisfaction rate through personalized dishes and on demand based pricing.• This will even increase the revenue for the restaurant.• This will even the operations for the restaurant efficient.
Github Link:	https://github.com/htmw/2024F-pace-super-giants/wiki

Team Agreement

Team Collaboration Agreement

Purpose:

This document outlines the expectations, roles, and code of conduct for our 5-member team and sets priority for a positive, collaborative, and productive work environment. It will create the basis on which we will interact with each other to reach our goals for the project.

Guiding Principles:

1. Respect and Communication

Respect: All ideas, opinions, and contributions of every team member shall be valued. We will maintain the atmosphere of professionalism and support.

Open Communication: There shall be proper, straight forward, and timely communication. Any team member can share whatever they think, feedback, or any concern that comes to their mind.

Constructive Feedback: Regular feedback and at all times for the betterment of each other.

2. Roles and Responsibilities

Specific roles within the team will be assigned to each member, based on individual strengths and experience.

The roles can change as time progresses, since project needs will change. Any role should be subject to discussion and agreed upon among the team members.

3. Meetings and Communication Channels

Weekly Meetings: Our team shall meet once a week every Saturday at 3 pm for a full project update.

Daily Sync: A light check-in every day on Slack at 10 pm to take a call on immediate progress and blockers of the day.

Agenda: An agenda for the meeting shall be shared at least 24 hours in advance of each meeting by the team lead so that everyone comes prepared to the meeting.

Attendance: All members shall be present in each and every meeting. In case a member needs to miss any meeting, the member needs to be intimate in advance. Also, the member needs to go through the summary of the meeting shared on Slack. **Primary Communication Channel:** Slack will serve as our main communication channel, where the flow of information, discussion, and notifications will be shared.

4. Decision-Making Process

Consensus-Based Decisions: Most decisions will be made through consensus as a means of keeping everyone's opinions heard.

Majority Vote: If we cannot reach a consensus, there will be a majority vote.

Documentation: Key decisions are to be documented down on Slack for reference and transparency.

5. Conflict Resolution

Open Discussions: Any kind of conflict or misunderstanding will be discussed openly and resolved respectfully among the team members.

6. Task Allocation and Management

Task Assignment: Tasks will be assigned by expertise and workload balance to ensure an equal amount of responsibilities.

Flexibility: Group members can seek help and/or redistribute the tasks among themselves if someone feels overwhelmed.

Tools: We will be using GitHub for version control and Slack for updating on task management.

7. Performance and Feedback

Weekly Performance Reviews: Individual and team performance will be reviewed each week in order to keep aligned with project goals.

Continuous Improvement: Regular sessions of constructive feedback will enable us to identify the problems at an early stage and fine tune our approach with time.

8. Change in Agreement

This is a living agreement: hence, the team has reserved the right to change it at any time if needed. Proposed changes will be discussed and mutually agreed upon by all parties concerned.

Team Members

1. Deekeshitha Navuluri
2. Bhavya sri Panguluri
3. Manoj kumar Ambavarapu
4. Sai mahesh Sandeboina
5. Phanindhr Thota



Sarah



Age: 29

Occupation: Fitness Instructor

Goals: Sarah is always looking for healthy meal options that fit her dietary restriction (gluten-free, low-carb) while exploring new dishes.



Frustrations: Most restaurant menus are not customized for her dietary needs, making it harder for her to find suitable dishes or restaurants.



How the app helps: The app provides personalized menu suggestions based on her dietary preferences and tracks her past orders, offering healthy options without the need for constant searching.



David



Age: 45

Occupation: Restaurant Owner

Goals: David wants to increase customer satisfaction and optimize pricing strategies to boost profitability during peak and off-peak hours.



Frustrations: Struggles with pricing static menus efficiently, as well as managing inventory and customer preferences.



How the app helps: The app uses dynamic pricing to adjust menu prices based on demand and time of day, while also offering personalized menu suggestions to customers, helping David increase sales and reduce food waste.



Emily

- ★ **Age:** 21
- ★ **Occupation:** College Student
- ★ **Goals:** Emily wants to enjoy meals at restaurants without exceeding her budget, especially during social outings with friends.
- ★ **Frustrations:** High prices during peak times make dining out expensive, and she struggles to find good deals or discounts.
- ★ **How the app helps:** The app provides Emily with meal recommendations based on her preferences and budget, while offering her dynamic discounts based on off-peak times or restaurant promotions.

MVP

★ User and Restaurant Onboarding

- **Customer and Restaurant Sign-up:** Email and social login (Google).
- **Basic Profile Setup:** Customers set dietary preferences, restaurants onboard basic menu.

★ Authentication & Login

- **Login/Sign-up:** Simple email/password or social login.
- **Session Management:** Basic secure login flow.

★ Personalized Menu Recommendations

- **ML-powered Recommendations:** Personalized dish suggestions based on customer preferences.

★ Dynamic Pricing

- **Real-Time Price Updates:** Dish prices adjust dynamically based on demand and time of day.

★ Basic Order Flow

- **Add to Cart & Checkout:** Users can select dishes, view cart, and place orders.



Technologies



Languages

- TypeScript
- Python
- NoSQL



Frameworks

- ReactJS
- Node JS
- Flask
- Pytorch



Tools

- Github
- MongoDB
- Trello
- Azure
- Slack

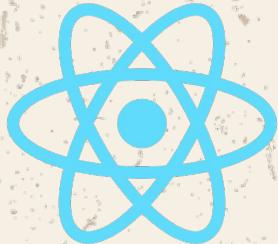
Languages



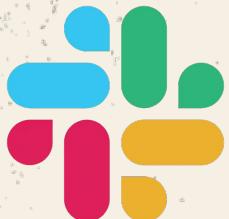
- ★ TypeScript is subset of Javascript with static typing, that makes the code easy to debug. It is useful in this project, as it code reliable and good for large scale projects and it's used for developing client side as well the server side of the application
- ★ Python is high level programming language and its known for its readability and for this project is it used for developing machine learning models and models API.
- ★ NoSQL is a database, where data storing is flexible. Schema less format like key value store. This is efficient in storing user profiles, past orders and many.

Frameworks

- ★ ReactJS its a Javascript library which is used for building UI and for single page applications.
- ★ NodeJS is a Javascript runtime which allows to build scalable applications and for this project it is used for handling APIs requests and managing connection between client and server side.
- ★ Flask is lightweight python framework and for this project is it used for building Machine learning API.
- ★ Pytorch is deep learning library and it is used for building deep learning models for this project.



Tools



- ★ Github is a platform used for version control and collaboration. In this project it is used for tracking the code and manage the code base.
- ★ MongoDB its a NOSql data and its used for storing the data.
- ★ Trello is project management tool using boards and cards and for this project it is used for track progress and managing tasks.
- ★ Azure is a cloud computing platform where the website services are hosted.
- ★ Slack is a communicable tools which is designed for teams and for this project, it is used for communication among team members.

Algorithms



Neural Collaborative Filtering (NCF)

This Algorithm is just combining the traditional collaboration approach with neural networks. NCF captures the complex patterns with users needs and how they interact based on that, it recommends the dishes.



Deep Reinforcement Learning (DRL)

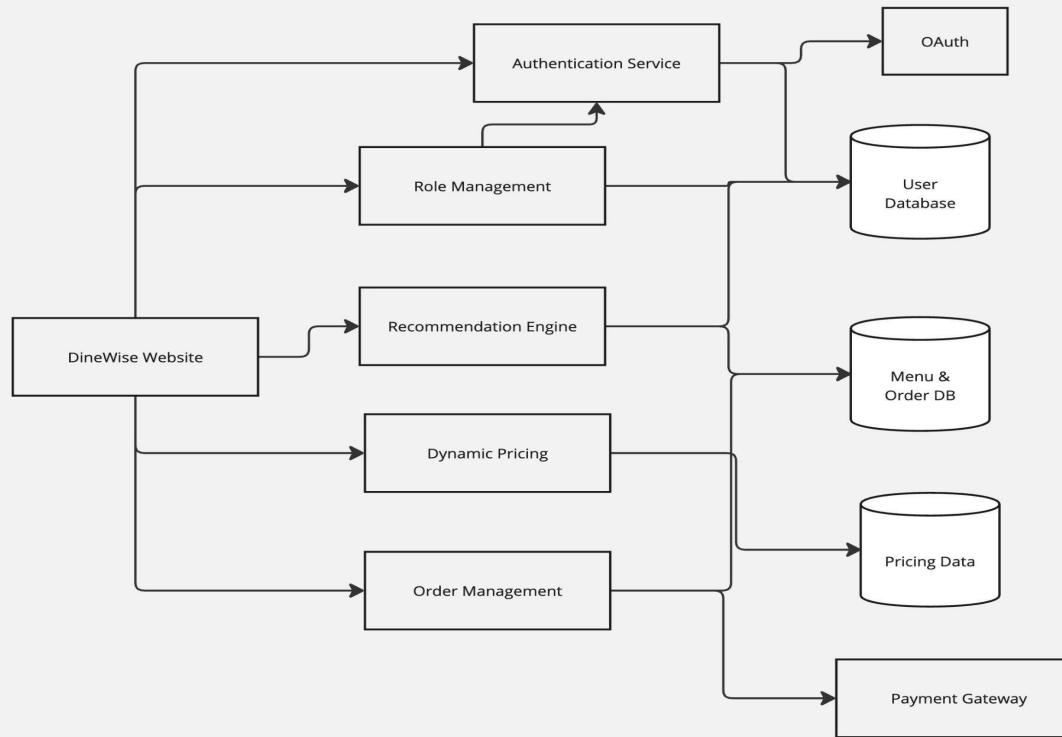
This Algorithm is just a combination of reinforcement learning with neural network. This helps the this system in dynamic pricing by learning it through trail and error.



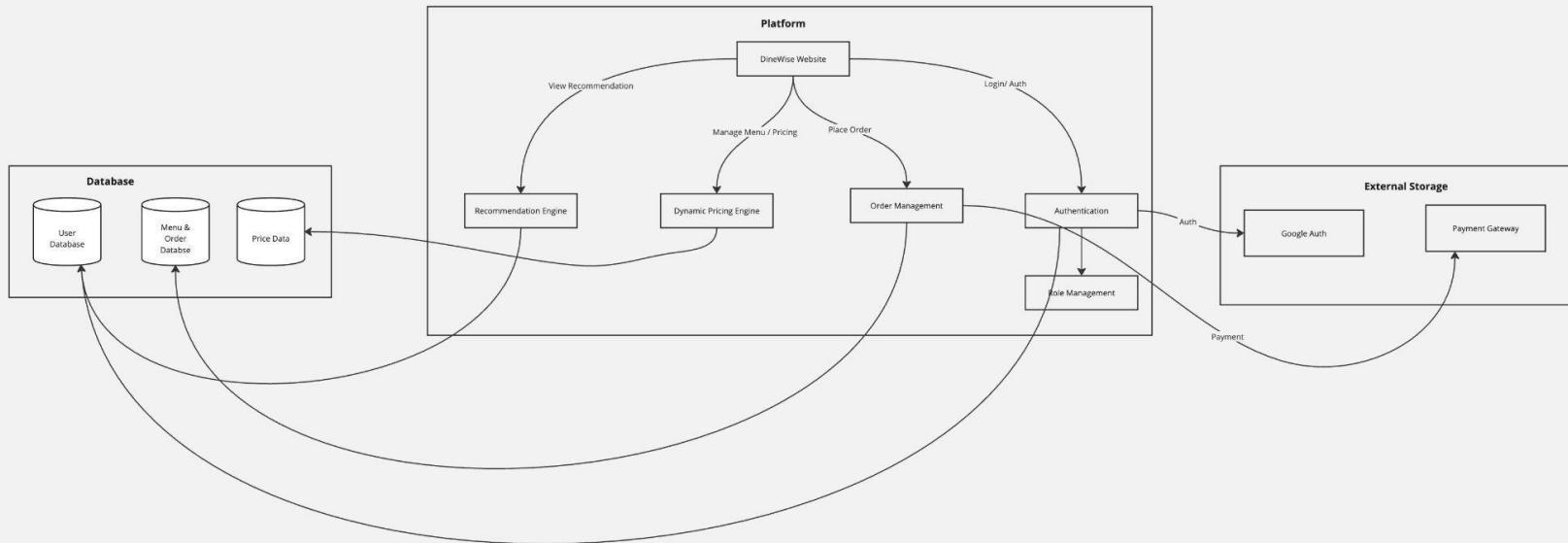
AutoEncoders

This Algorithm is used unsupervised for unsupervised learning approach and it is used for understanding the dishes content based on their ingredients.

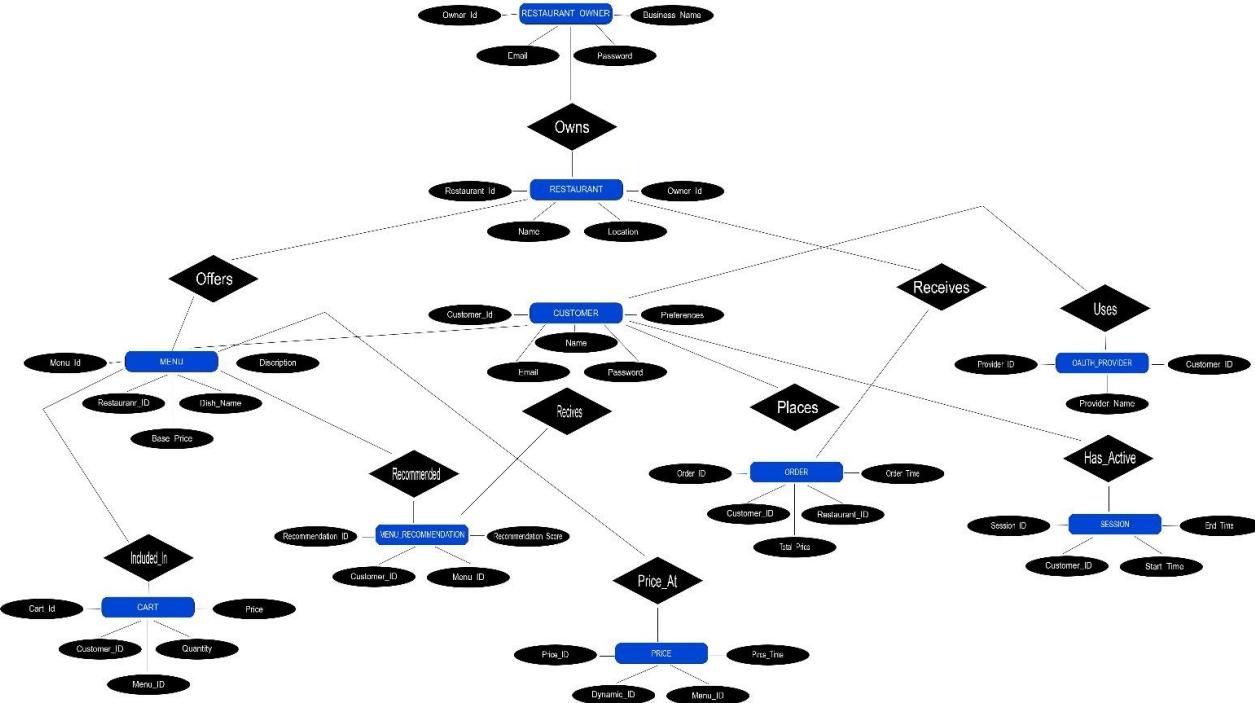
Architecture Diagram



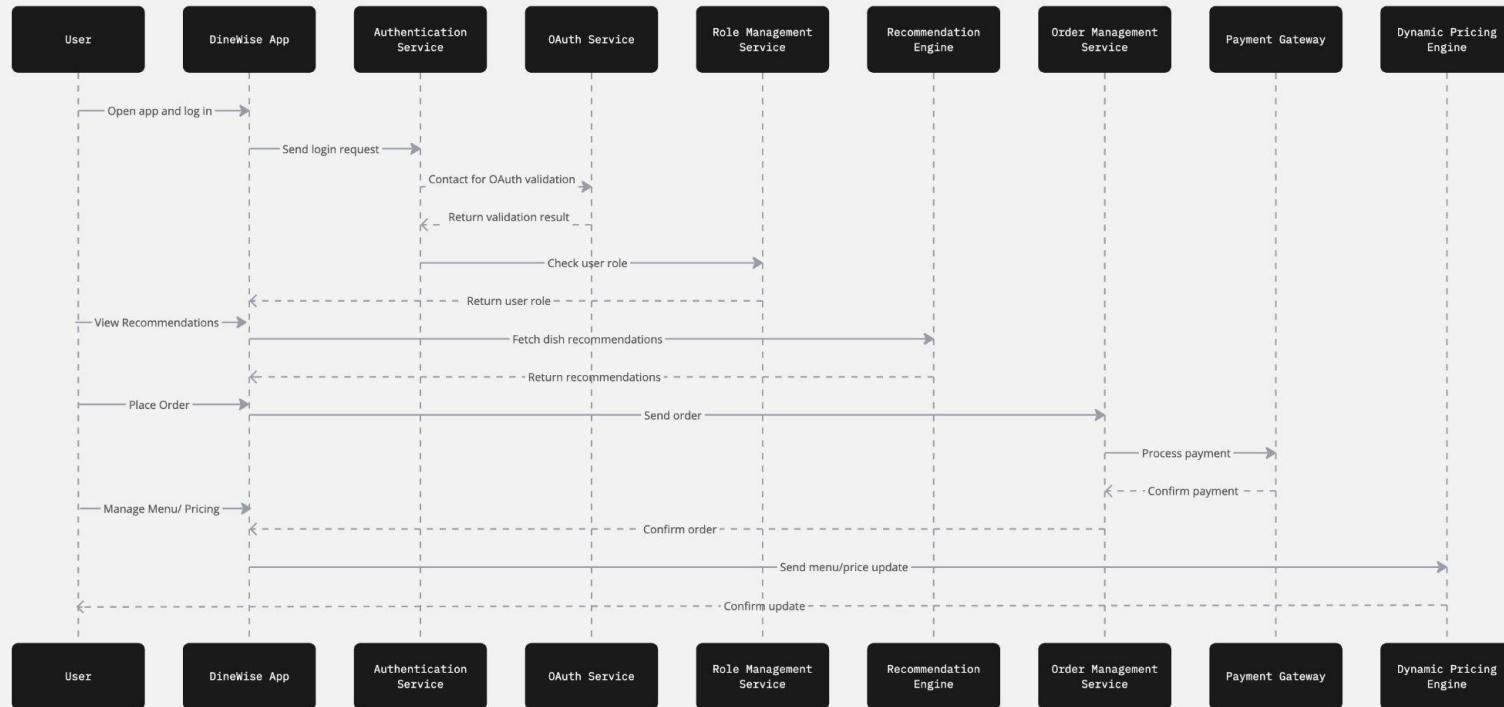
Context Diagram



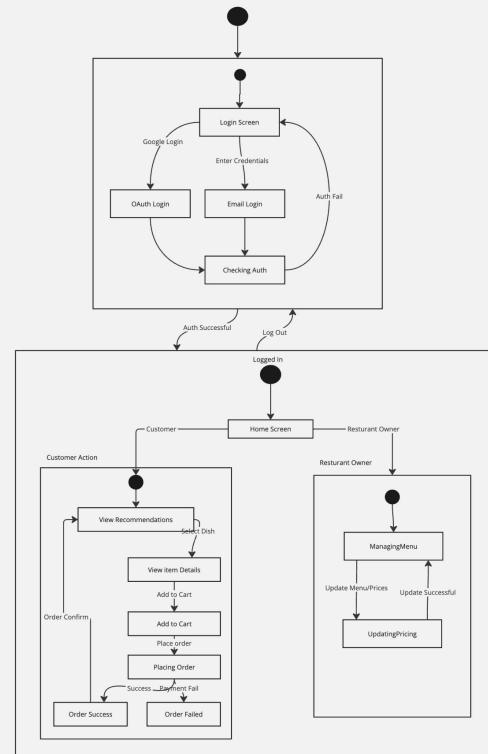
ER Diagram



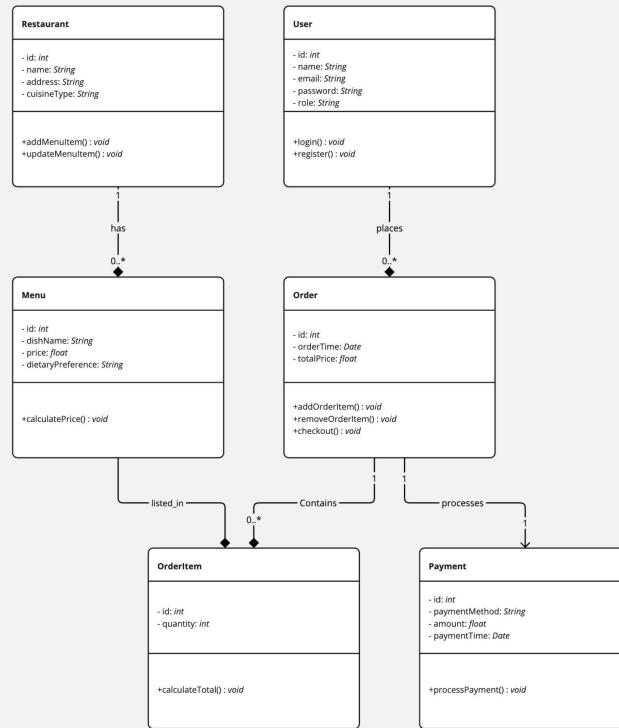
Sequence Diagram



State Diagram



Class Diagram



Sprint 1 Recap

we completed the core functionalities: Customer and Restaurant Registration, along with Login for both customers and restaurant owners.

Product Backlog

ID	User Story / Task	Description	Priority	Acceptance Criteria	Sprint	User Benefits
US-01	User Story: Customer Registration	As a customer, I want to register and create a profile to receive personalized menu recommendations.	High	- Users can sign up via email or social login. - Successful sign-up redirects to profile setup.	Sprint 1	Enables customers to register quickly and receive tailored menu suggestions.
US-02	User Story: Restaurant Registration	As a restaurant owner, I want to create an account to list my restaurant and its menu on DineWise.	High	- Owners can sign up using business credentials. - Successful sign-up redirects to the dashboard.	Sprint 1	Allows restaurant owners to list their restaurant and expand their customer reach.
US-03	User Story: Customer Login	As a customer, I want to log in to my account using registered credentials to access my personalized recommendations.	High	- Users can log in via email/password. - Successful login redirects to the dashboard.	Sprint 1	Enables customers to log in and view personalized recommendations based on their preferences.

Product Backlog

ID	User Story / Task	Description	Priority	Acceptance Criteria	Sprint	User Benefits
US-04	User Story: Restaurant Owner Login	As a restaurant owner, I want to log in to manage my restaurant menu and customer activity.	High	- Owners can log in with email/password. - Successful login redirects to the restaurant dashboard.	Sprint 1	Empowers owners to manage their restaurant details, menu, and customer interactions effectively.
US-05	User Story: Password Recovery	As a user, I want to recover my password if I forget it.	Medium	- Users can request a password reset link. - Successful password reset allows a new login.	Sprint 1	Provides users with a secure and convenient way to recover account access.
US-06	User Story: Enable OAuth Login	As a user, I want to log in using Google for easy access.	Medium	- Users can log in with Google. - OAuth login redirects users back to the app.	Sprint 1	Simplifies the login process and improves user experience with multiple login options.

Product Backlog

ID	User Story / Task	Description	Priority	Acceptance Criteria	Sprint	User Benefits
TS-01	Technical Task: Session Management	Implement secure session management for logged-in users.	High	<ul style="list-style-type: none">- Sessions expire after inactivity.- Users remain logged in until they log out manually.	Sprint 1	Enhances security by managing user sessions and protecting sensitive information.
US-07	User Story: Personalized Menu Suggestions	As a customer, I want to receive menu recommendations based on my preferences and dining history.	High	<ul style="list-style-type: none">- Menu recommendations are personalized.- Menu updates dynamically based on preferences.	Sprint 2	Provides a tailored dining experience, making it easier for customers to choose their meals.
US-08	User Story: Add Items to Cart	As a customer, I want to add items to my cart so I can place an order from my personalized menu.	Medium	<ul style="list-style-type: none">- Customers can add items to their cart.- Cart updates in real time with selected dishes.	Sprint 2	Allows customers to easily manage items for their order, improving the purchasing experience.

Product Backlog

ID	User Story / Task	Description	Priority	Acceptance Criteria	Sprint	User Benefits
TS-02	Technical Task: Menu Recommendation Algorithm	Develop the ML model for personalized dish recommendations based on user preferences.	High	<ul style="list-style-type: none">- Recommendation engine uses customer data.- Personalized suggestions are ranked accurately.	Sprint 2	Increases customer satisfaction by ensuring accurate and relevant menu suggestions.
TS-03	Technical Task: Data Pipeline for ML	Set up a data pipeline to collect customer preferences and feedback for the recommendation model.	High	<ul style="list-style-type: none">- Customer data is stored and pre-processed.- Data is fed into the ML model for training.	Sprint 2	Ensures the recommendation model is updated with current user preferences, improving its accuracy.
TS-04	Technical Task: Real-Time Pricing Backend	Create backend infrastructure for real-time dynamic pricing updates based on demand.	High	<ul style="list-style-type: none">- Backend handles real-time price updates.- Pricing changes are served to the frontend API.	Sprint 2	Allows for flexible and dynamic pricing, enhancing customer satisfaction and profitability.

Product Backlog

ID	User Story / Task	Description	Priority	Acceptance Criteria	Sprint	User Benefits
US-09	User Story: Dynamic Pricing Display	As a customer, I want to see the price of each dish updated in real-time based on demand.	High	<ul style="list-style-type: none">- Prices adjust dynamically in real-time.- Users see updated prices on the menu.	Sprint 3	Keeps customers informed about current pricing, improving transparency and decision-making.
US-10	User Story: Remove Items from Cart	As a customer, I want to remove item in my cart before placing the order.	Medium	<ul style="list-style-type: none">- Customers can remove items.- Cart updates immediately with accurate pricing.	Sprint 3	Empowers customers to make adjustments to their order, enhancing flexibility and satisfaction.
US-11	User Story: View Cart Summary	As a customer, I want to view a summary of my cart before checkout to confirm order details.	Medium	<ul style="list-style-type: none">- Cart displays selected items and total cost.- Cart updates dynamically with pricing.	Sprint 3	Allows customers to review and confirm their order, ensuring accuracy before payment.

Product Backlog

ID	User Story / Task	Description	Priority	Acceptance Criteria	Sprint	User Benefits
US-12	User Story: Place Order	As a customer, I want to place an order for the recommended dishes.	High	<ul style="list-style-type: none">Customers can place an order after adding items to the cart.Successful order redirects to confirmation.	Sprint 3	Simplifies the ordering process, allowing customers to place orders quickly and conveniently.
TS-05	Technical Task: Dynamic Pricing Algorithm	Develop the dynamic pricing algorithm that adjusts prices based on demand and availability.	High	<ul style="list-style-type: none">Pricing algorithm adjusts in real-time.Customers see price changes in real time.	Sprint 3	Provides an adaptive pricing model, ensuring competitive and fair pricing for customers.

Sprint 1 (Carried Forward) Backlog

ID	User Story / Task	Description	Story Points	Acceptance Criteria
US-05	User Story: Password Recovery	As a user, I want to recover my password if I forget it.	3	<ol style="list-style-type: none">Users can request a password reset link.Successful password reset allows a new login.
US-06	User Story: Enable OAuth Login	As a user, I want to log in using Google for easy access.	5	<ol style="list-style-type: none">Users can log in with Google.OAuth login redirects users back to the app.
TS-01	Technical Task: Session Management	Implement secure session management for logged-in users.	3	<ol style="list-style-type: none">Sessions expire after inactivity.Users remain logged in until they log out manually.

Sprint 2 Backlog

ID	Story/Task	Description	Story Points	Acceptance Criteria
US-07	User Story: Personalized Menu Suggestions	As a customer, I want to receive menu recommendations based on my preferences and dining history.	8	<ol style="list-style-type: none">Menu recommendations are personalized.Menu updates dynamically based on preferences.
US-08	User Story: Add Items to Cart	As a customer, I want to add items to my cart so I can place an order from my personalized menu.	5	<ol style="list-style-type: none">Customers can add items to their cart.Cart updates in real-time with selected dishes.
TS-02	Technical Task: Menu Recommendation Algorithm	Develop and integrate the machine learning model for personalized dish recommendations.	8	<ol style="list-style-type: none">Recommendation engine uses customer data.Personalized suggestions are ranked accurately.
TS-03	Technical Task: Data Pipeline for ML	Set up a data pipeline to collect customer preferences and feedback for the recommendation model.	5	<ol style="list-style-type: none">Customer data is stored and pre-processed.Data is fed into the ML model for training.
TS-04	Technical Task: Real-Time Pricing Backend	Create backend infrastructure for real-time dynamic pricing updates based on demand.	4	<ol style="list-style-type: none">Backend handles real-time price updates.Pricing changes are served to the frontend API.

Test Cases

ID	User Story / Task	Test Case	Expected Result	Actual Result
TC-01	Password Recovery (US-05)	Test password reset request with a valid email.	User receives a password reset link via email.	Completed as expected.
TC-02	Password Recovery (US-05)	Test reset link functionality and creation of new password.	User can set a new password after clicking the reset link.	Completed as expected.
TC-03	Enable OAuth Login (US-06)	Test Google login flow.	Google login redirects the user to the dashboard.	Completed as expected.
TC-04	Session Management (TS-01)	Test session expiry after the defined period of inactivity.	Session automatically expires after a set time (e.g., 30 minutes) of inactivity.	Completed as expected.

Test Cases

ID	User Story / Task	Test Case	Expected Result	Actual Result
TC-05	Session Management (TS-01)	Test that users remain logged in until they manually log out.	Users remain logged in unless they manually log out.	Completed as expected.
TC-06	Personalized Menu Suggestions (US-07)	Verify personalized menu recommendations based on customer profile.	Personalized recommendations displayed based on user preferences.	Completed as expected.
TC-07	Personalized Menu Suggestions (US-07)	Test real-time updates to recommendations with changing preferences.	Menu updates dynamically in response to changes in preferences.	Completed as expected.
TC-08	Menu Recommendation Algorithm (TS-02)	Validate recommendation accuracy based on initial user preferences.	Recommendations align with the user's profile and history.	Completed as expected.
TC-09	Menu Recommendation Algorithm (TS-02)	Validate recommendation ranking accuracy.	High relevance of recommended items at the top.	Completed as expected.

Test Cases

ID	User Story / Task	Test Case	Expected Result	Actual Result
TC-10	Data Pipeline for ML (TS-03)	Validate data collection and storage of user preferences.	Data is pre-processed and stored correctly.	Completed as expected.
TC-11	Data Pipeline for ML (TS-03)	Verify the pipeline's integration with the recommendation engine.	Data is successfully passed to the model for training.	Completed as expected.
TC-12	Add Items to Cart (US-08)	Test adding items to cart and real-time update.	Cart updates in real-time with selected dishes.	Not completed (issue with real-time update).
TC-13	Add Items to Cart (US-08)	Validate multiple items can be added.	Multiple items added without errors.	Not completed (issue with adding multiple items).
TC-14	Real-Time Pricing Backend (TS-04)	Verify backend handling for real-time price updates based on demand.	Real-time pricing updates served to frontend API.	Not completed (API delays).

Test Cases

ID	User Story / Task	Test Case	Expected Result	Actual Result
TC-15	Dynamic Pricing Display (US-09)	Test real-time display of dynamic prices on customer menu.	Prices adjust dynamically in real-time as demand changes.	Not completed (Carried to Sprint 3).

Sprint 2 Completed

ID	User Story / Task	Status
US-05	Password Recovery	Completed in Sprint 2
US-06	Enable OAuth Login	Completed in Sprint 2
TS-01	Session Management	Completed in Sprint 2
US-07	Personalized Menu Suggestions	Completed in Sprint 2
TS-02	Menu Recommendation Algorithm	Completed in Sprint 2
TS-03	Data Pipeline for ML	Completed in Sprint 2

Sprint 2 Not Completed

ID	User Story / Task	Status	Reason for Incompletion
US-08	Add Items to Cart	Not Completed (Carried to Sprint 3)	Issue with real-time update and adding multiple items
TS-04	Real-Time Pricing Backend	Not Completed (Carried to Sprint 3)	API delays impacting

Metrics

Team Velocity - Sprint 2

- **Total Story Points Committed:** 41 story points
- **Total Story Points Completed:** 32 story points

Completed/Committed Ratio

- **Committed Story Points:** 41 story points
- **Completed Story Points:** 32 story points

Thus, the **Completed/Committed Ratio for Sprint 2 is 78.05%**.

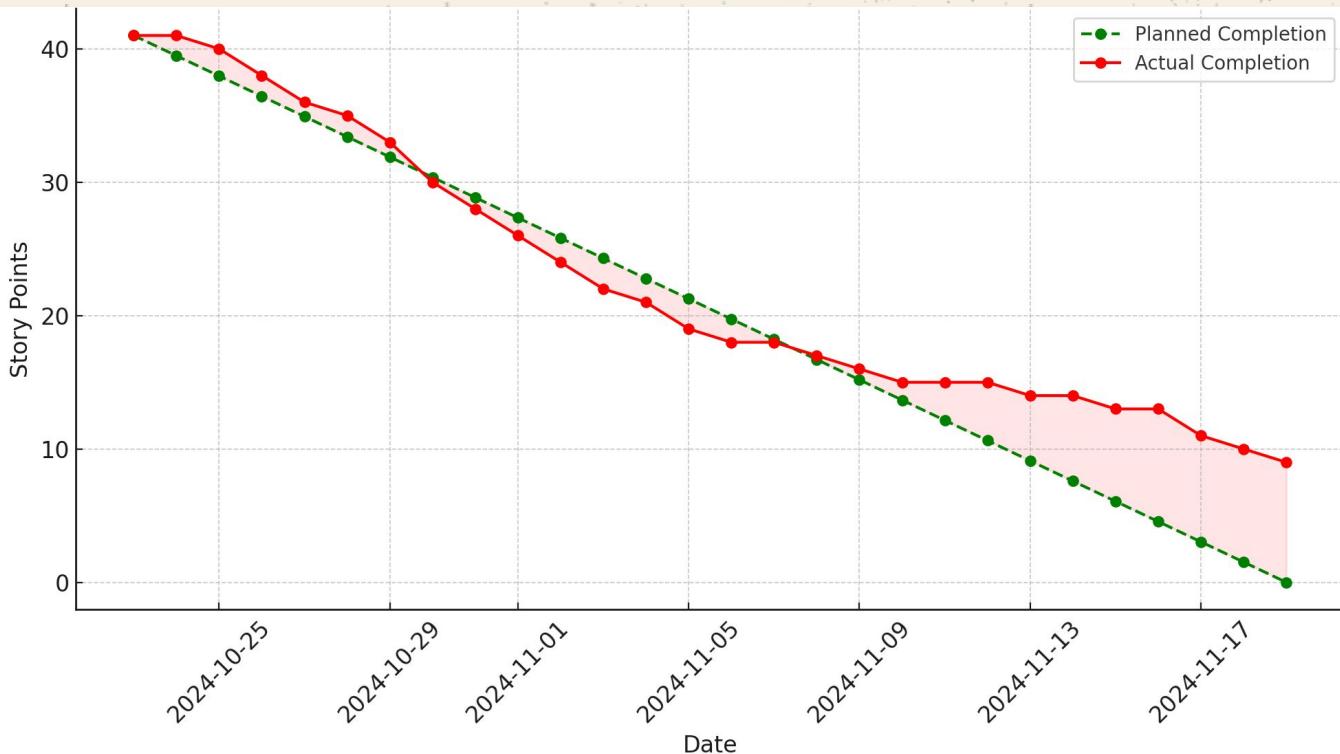
Average Velocity (Sprint 1 and Sprint 2)

Sprint	Committed Story Points	Completed Story Points	Completed/Committed Ratio
Sprint 1	27	16	59.26%
Sprint 2	41	32	78.05%

Average Completed Story Points: 24 story points

Average Completed/Committed Ratio: 68.65%

Burndown Chart



Retrospective

Sprint 2

What went well

good pairing work was done deposit remote work	efficient work flow
+1	+1
Good Team collaboration and work	willing to work on new tech stack
+0	+0
giving regular update about assign work	
+0	

What can be improved

clear idea about the given task and also we need to spend more time on code	before picking any story each story should have clear acceptance criteria
+1	+1
Need to have more team build events	Need to update the story board on daily basis
+0	+0

Action Items

To improve the story board update, Need to discuss on daily basis to update the story board daily according to the work completed.	guarantee all stories have clear acceptance criteria before picking them up
+2	+1
Taking more time for technical part	To improve or idea about task ,After picking up a story need to analyse and divide into different tasks based on priority and requirement
+0	+0

Sprint 3

ID	Story/Task	Description	Story Points	Acceptance Criteria
US-09	User Story: Dynamic Pricing Display	As a customer, I want to see the price of each dish updated in real-time based on demand.	4	<ul style="list-style-type: none">- Prices adjust dynamically in real-time.- Users see updated prices on the menu.
US-10	User Story: Remove Items from Cart	As a customer, I want to remove or change items in my cart before placing the order.	3	<ul style="list-style-type: none">- Customers can remove or modify items.- Cart updates immediately with accurate pricing.
US-11	User Story: View Cart Summary	As a customer, I want to view a summary of my cart before checkout to confirm order details.	3	<ul style="list-style-type: none">- Cart displays selected items and total cost.- Cart updates dynamically with pricing.
US-12	User Story: Place Order	As a customer, I want to place an order for the recommended dishes.	5	<ul style="list-style-type: none">- Customers can place an order after adding items to the cart.- Successful order redirects to confirmation.

Sprint 3

ID	Story/Task	Description	Story Points	Acceptance Criteria
TS-05	Technical Task: Dynamic Pricing Algorithm	Develop the dynamic pricing algorithm that adjusts prices based on demand and availability.	5	<ul style="list-style-type: none">- Pricing algorithm adjusts in real-time.- Customers see price changes in real-time.
US-08	User Story: Add Items to Cart	As a customer, I want to add items to my cart so I can place an order from my personalized menu.	5	<ul style="list-style-type: none">- Customers can add items to their cart.- Cart updates in real-time with selected dishes.
TS-04	Technical Task: Real-Time Pricing Backend	Create backend infrastructure for real-time dynamic pricing updates based on demand.	4	<ul style="list-style-type: none">- Backend handles real-time price updates.- Pricing changes are served to the frontend API.

Website Screenshots

DineWise

[Home](#)

[Register](#)

[Login](#)

Welcome to DineWise

Experience personalized dining with AI-driven menu
recommendations and dynamic pricing

[Get Started](#)

Website Screenshots

Customer Restaurant

First Name Last Name

Phone Number +1 (234) 567-8900

Email your@email.com

Password Min. 8 characters

Confirm Password Confirm your password

By registering, you agree to our [Terms of Service](#) and [Privacy Policy](#)

Register as Customer

Website Screenshots

Customer Restaurant

Business Name

Business Address

Business Phone

+1 (234) 567-8900

Email

your@email.com

Password

Min. 8 characters

Confirm Password

Confirm your password

Website Screenshots

Welcome Back

Sign in to your DineWise account

[Customer](#) [Restaurant](#)

Email
 [✉](#)

Password
 [👁](#)

Remember me [Forgot password?](#)

[Sign In](#)

Or continue with

 Continue with Google

Don't have an account? [Sign up here](#)

Website Screenshots

Welcome Back

Sign in to your restaurant dashboard

[Customer](#) [Restaurant](#)

Email
 [✉](#)

Password
 [ⓘ](#)

Remember me [Forgot password?](#)

[Sign In](#)

Don't have an account? [Sign up here](#)

Website Screenshots

DineWise Restaurant Dashboard

phani →



Menu Management

+ Add Menu Item

Search menu items...

All Categories

ITEM	CATEGORY	PRICE	STATUS	ACTIONS
Crispy Spring Rolls Hand-rolled crispy rolls filled with vegetables, glass noodles, and mushrooms. Served with sweet chili sauce.	Appetizers	\$7.99	active	

Website Screenshots

The screenshot displays the DineWise Restaurant Dashboard. At the top, there's a header with the logo "DineWise" and the text "Restaurant Dashboard". On the right side of the header are icons for notifications, settings, and user authentication (username "phani" and a log-out arrow).

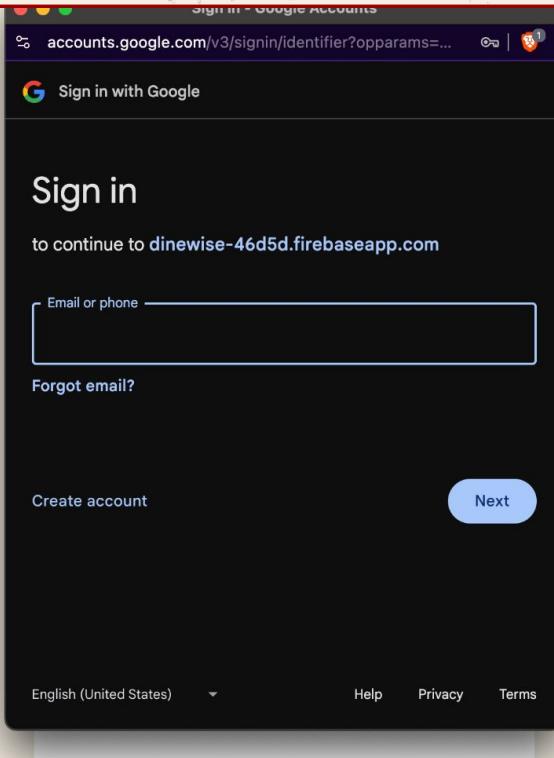
The main dashboard features several key metrics: "Total Revenue \$0.00" with a bar chart icon, "Avg. Order Value \$0.00" with a dollar sign icon, and a search bar labeled "Search menu items...". Below these is a section titled "Menu Management" with a "Search menu items..." input field and a table listing menu items.

The table in "Menu Management" has columns for "ITEM", "DESCRIPTION", "PRICE", "CATEGORY", and "PREPARATION TIME (MINUTES)". One item listed is "Crispy Spring Rolls" with the description "Hand-rolled crispy rolls filled with veget".

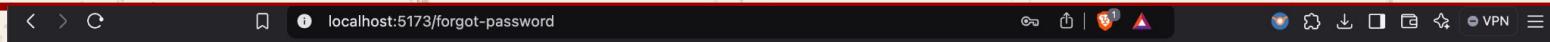
A central modal window titled "Add New Menu Item" is open. It contains fields for "Name" (with a placeholder "Enter name"), "Description" (with a placeholder "Enter description"), "Price" (\$), "Category" (with a dropdown menu showing "Select Category"), "Preparation Time (minutes)" (with a placeholder "Enter preparation time"), and "Dietary Restrictions" (with checkboxes for Vegetarian, Gluten-Free, Nut-Free, Kosher, Vegan, Dairy-Free, and Halal). At the bottom of the modal are "Cancel" and "Save Item" buttons.

The overall interface is clean and modern, using a light gray background with red and blue accents for buttons and highlights.

Website Screenshots



Website Screenshots



Reset Password

Enter your email to reset your password

Email

[Continue](#)

[← Back to login](#)

Website Screenshots

The screenshot shows a web browser window with a dark header bar. The address bar displays "localhost:5173/Udashboard". The main content area is titled "DineWise" in red. A "Welcome back!" message is displayed above a search bar containing "Search restaurants...". To the right of the search bar is a red button labeled "Update Preferences". Below the search bar is a row of cuisine filters: "All" (highlighted in red), Indian, Italian, Chinese, Mexican, Japanese, and American. Three restaurant cards are listed below the filters:

- bhavya** (star icon)
Various Cuisine
📍 2728 john f kennady ,blvd
⌚ 15714360352
[View Menu >](#)
- deekshitha** (star icon)
Various Cuisine
📍 2728 john f kennady ,blvd
⌚ 15714360352
[View Menu >](#)
- phani** (star icon)
Various Cuisine
📍 215 linden ave
⌚ 83748928787
[View Menu >](#)

Website Screenshots

The screenshot shows a web browser window with a dark header bar. The address bar displays "localhost:5173/Udashboard". The main content area is titled "DineWise" in red. A "Welcome back!" message is displayed above a search bar containing "Search restaurants...". To the right of the search bar is a red button labeled "Update Preferences". Below the search bar is a row of cuisine filters: "All" (highlighted in red), Indian, Italian, Chinese, Mexican, Japanese, and American. Three restaurant cards are listed below the filters:

- bhavya** (star icon)
Various Cuisine
📍 2728 john f kennady ,blvd
⌚ 15714360352
[View Menu >](#)
- deekshitha** (star icon)
Various Cuisine
📍 2728 john f kennady ,blvd
⌚ 15714360352
[View Menu >](#)
- phani** (star icon)
Various Cuisine
📍 215 linden ave
⌚ 83748928787
[View Menu >](#)

Website Screenshots

A screenshot of a web browser window displaying a form for selecting dietary restrictions. The URL in the address bar is `localhost:5173/preferences`. The page title is "Any dietary restrictions?". A sub-instruction says "Select all that apply". There are five input fields: "Vegetarian" (radio button checked), "Vegan" (radio button unselected), "Halal" (checkbox unselected), "Kosher" (checkbox unselected), and "No restrictions" (checkbox unselected). Navigation buttons "Previous" and "Next" are at the bottom.

localhost:5173/preferences

Back to Dashboard

Any dietary restrictions?

Select all that apply

Vegetarian

Vegan

Halal

Kosher

No restrictions

Previous

Next

API

The screenshot shows the Postman application interface. The top navigation bar includes links for Home, Workspaces, and API Network, along with search and upgrade buttons. The main workspace is titled "Team Workspace" and contains sections for Collections, Environments, and History. A collection named "API documentation" is selected.

The central area displays a POST request to <http://127.0.0.1:5000/recommend>. The request details show the method as POST, the URL, and various tabs for Params, Authorization, Headers, Body, Pre-request Script, Tests, and Settings. The Body tab is currently active, showing a JSON response with three recommendations:

```
[{"recommendations": [{"category": "Italian", "description": "Fresh vegetables with gluten-free pasta", "features": "Pasta Primavera Fresh vegetables with gluten-free pasta Italian vegetarian", "id": "1", "is_spicy": false, "is_vegetarian": true, "name": "Pasta Primavera"}, {"category": "Italian", "description": "Classic pizza with gluten-free crust and fresh basil", "features": "Margherita Pizza Classic pizza with gluten-free crust and fresh basil Italian vegetarian", "id": "3", "is_spicy": false, "is_vegetarian": true, "name": "Margherita Pizza"}, {"category": "Indian", "description": "Mild curry with seasonal vegetables", "features": "Vegetable Curry Mild curry with seasonal vegetables Indian vegetarian", "id": "5", "is_spicy": false, "is_vegetarian": true, "name": "Vegetable Curry"}]}
```

The status bar at the bottom indicates a 200 OK response with a time of 32 ms and a size of 1.71 KB. The bottom navigation bar includes links for Online, Find and replace, Console, Postbot, Runner, Start Proxy, Cookies, Vault, Trash, and Help.

Wikipage Link

<https://github.com/htmw/2024F-pace-super-giants/wiki>

Application Demo



Thank You

DineWise