



DineWise



Team 2 Pace Super Giants



TABLE OF CONTENTS



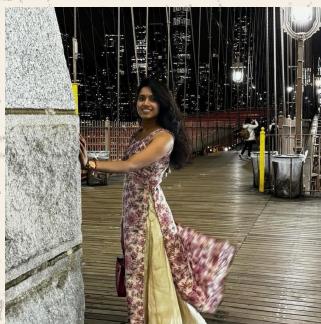
- **Team Member Roles and Responsibilities**
- **Improvements Made from Professor Feedback**
- **Personas**
- **Minimum Viable Product (MVP)**
- **Technologies**
- **Algorithms**
- **Diagrams**
 - Architecture Diagram
 - Context Diagram
 - ER Diagram
 - Sequence Diagram
 - State Diagram
 - Class Diagram
- **Product Backlog**
- **Sprint 3 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**
- **Project Demo - Sprint 3**
 - App Screenshot
 - API Slides
- **GitHub Link**
- **Live Application Demo**

OUR TEAM



★ **Deekshitha
Navuluri** ★

Full Stack Developer



★ **Panguluri
Bhavya** ★

Product Designer /
Frontend Developer



★ **Phanindhr
Thota** ★

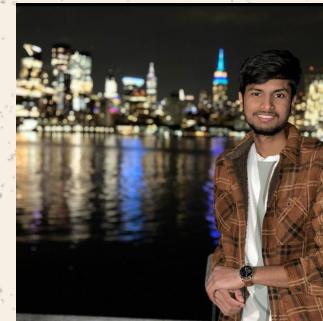
Frontend Developer

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

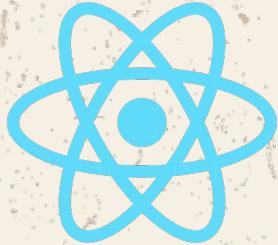
Languages



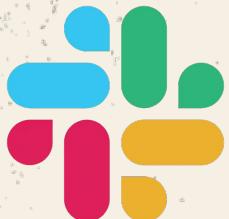
- ★ TypeScript is subset of Javascript with static typing, that makes the code easy to debug. It is useful in this project, as it code reliable and good for large scale projects and it's used for developing client side as well the server side of the application
- ★ Python is high level programming language 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)

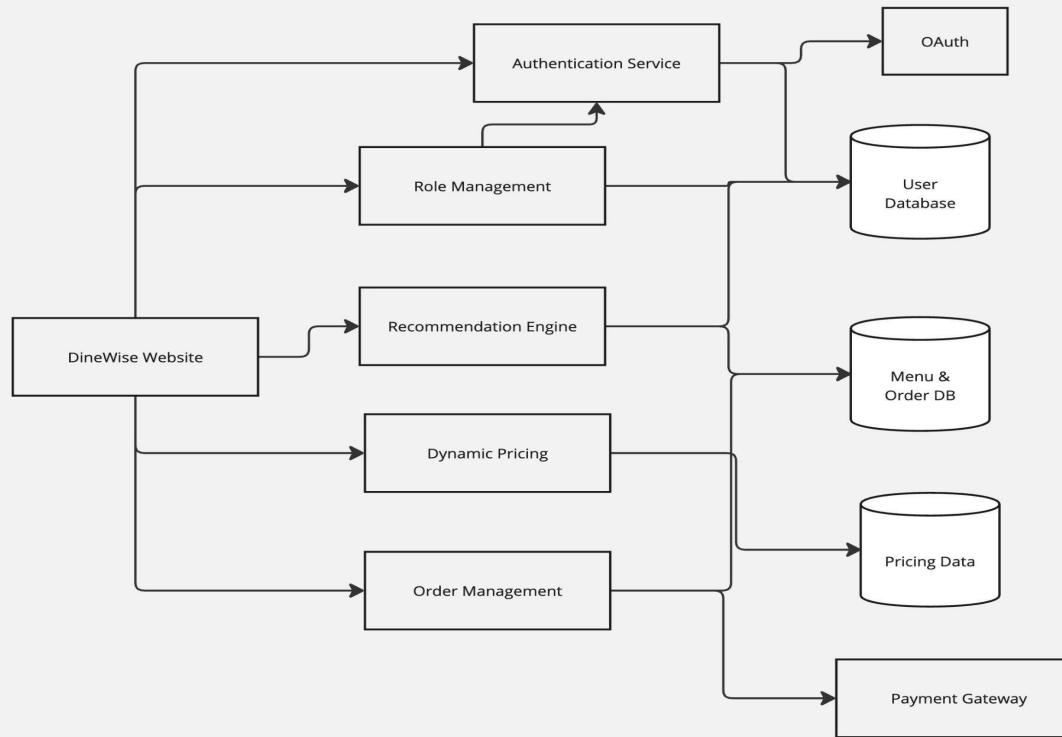
This Algorithm is just a combination of reinforcement learning with neural network. This helps the this system in dynamic pricing by learning it through trail and error.



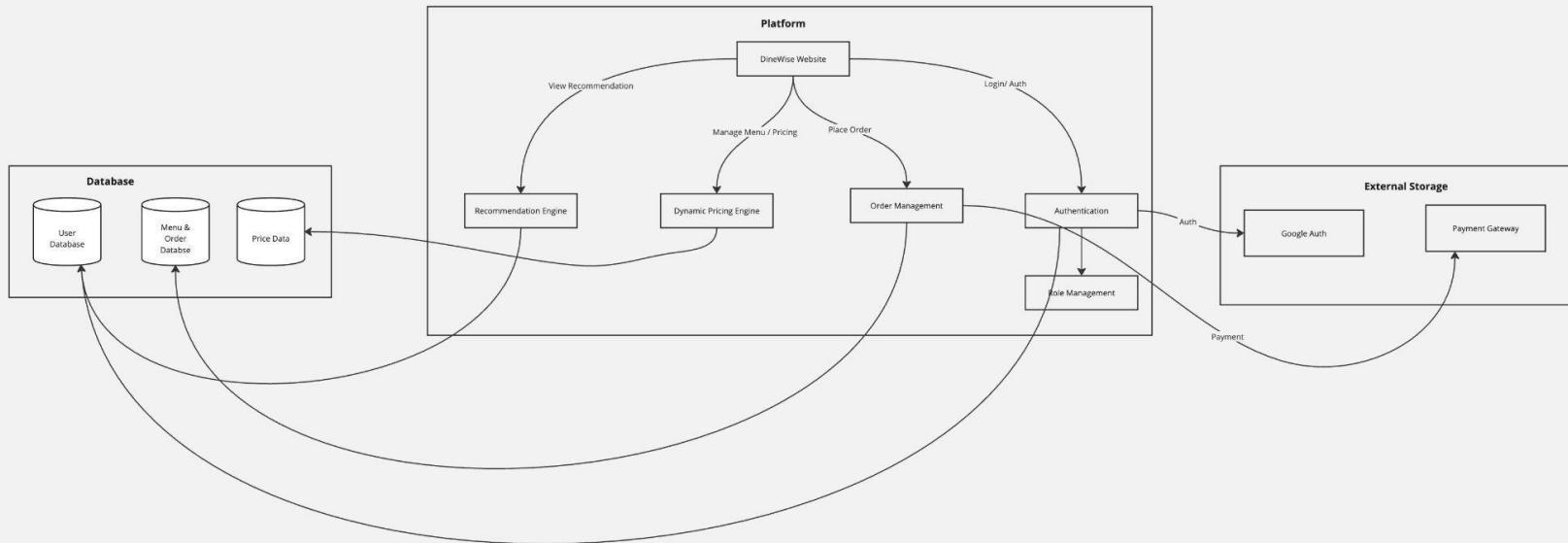
AutoEncoders

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

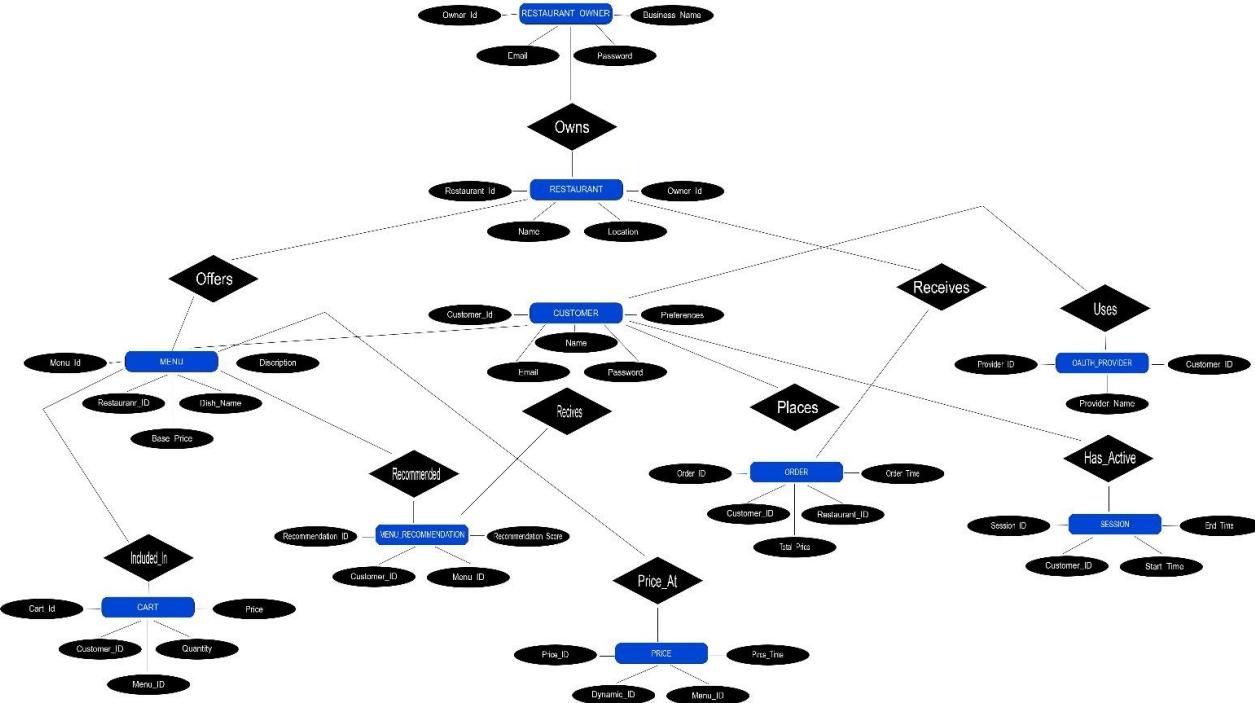
Architecture Diagram



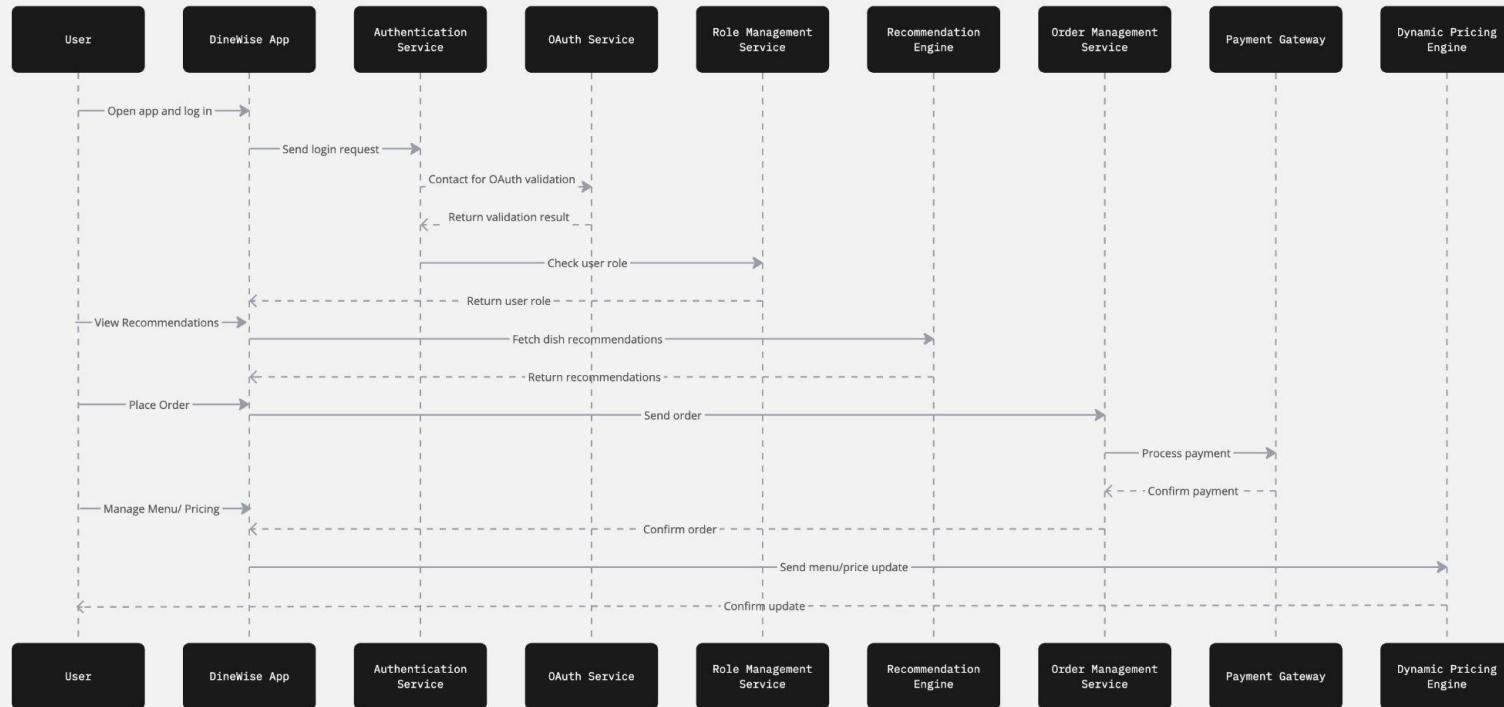
Context Diagram



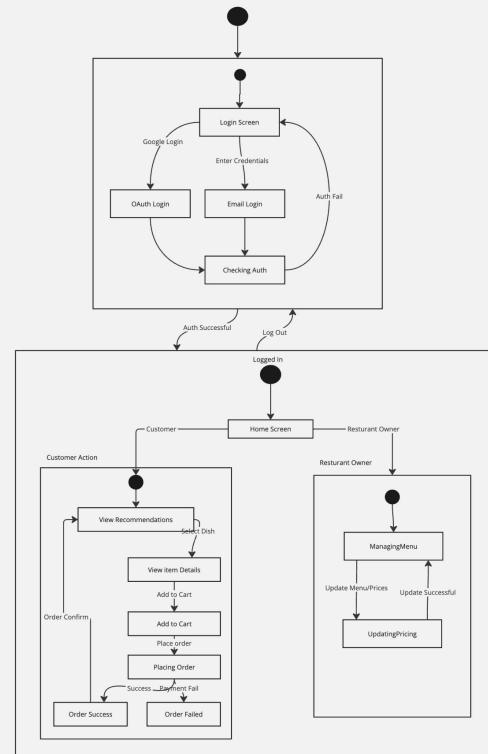
ER Diagram



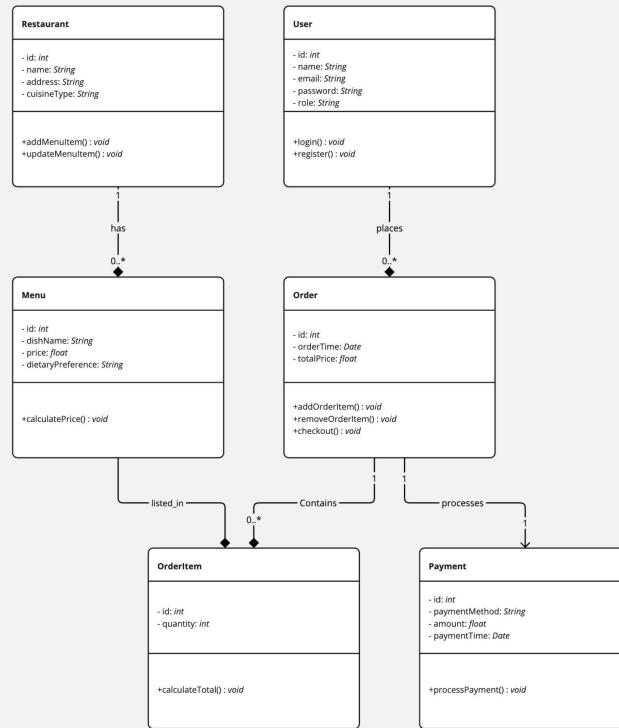
Sequence Diagram



State Diagram



Class Diagram



Product Backlog

ID	Story	Priority	Acceptance Criteria	Sprint
US-01	As a customer, I want to register and create a profile so that I can quickly sign up and receive tailored menu recommendations. so that I can quickly sign up and receive tailored menu recommendations.	High	- Users can sign up via email or social login. - Successful sign-up redirects to profile setup.	Sprint 1
US-02	As a restaurant owner, I want to create an account and list my restaurant and menu so that I can expand my customer reach. so that I can expand my customer reach.	High	- Owners can sign up using business credentials. - Successful sign-up redirects to the dashboard.	Sprint 1
US-03	As a customer, I want to log in using registered credentials so that I can access personalized recommendations conveniently. so that I can access personalized recommendations conveniently.	High	- Users can log in via email/password. - Successful login redirects to the dashboard.	Sprint 1
US-04	As a restaurant owner, I want to log in to manage my menu and customer activity so that I can efficiently handle operations. so that I can efficiently handle operations.	High	- Owners can log in with email/password. - Successful login redirects to the restaurant dashboard.	Sprint 1

Product Backlog

ID	Story	Priority	Acceptance Criteria	Sprint
US-05	As a customer, I want to recover my password so that I can regain access to my account securely. so that I can regain access to my account securely.	Medium	- Users can request a password reset link. - Successful password reset allows a new login.	Sprint 1
US-06	As a customer, I want to log in using Google so that I can access my account easily. so that I can access my account easily.	Medium	- Users can log in with Google. - OAuth login redirects users back to the app.	Sprint 1
TS-01	Implement secure session management for logged-in users so that sensitive information is protected during user sessions.	High	- Sessions expire after inactivity. - Users remain logged in until they log out manually.	Sprint 1
US-07	As a customer, I want to receive menu recommendations based on my preferences so that I can have a tailored dining experience. so that I can have a tailored dining experience.	High	- Menu recommendations are personalized. - Menu updates dynamically based on preferences.	Sprint 2

Product Backlog

ID	Story	Priority	Acceptance Criteria	Sprint
US-08	As a customer, I want to add items to my cart so that I can easily manage my order. so that I can easily manage my order.	Medium	- Customers can add items to their cart. - Cart updates in real time with selected dishes.	Sprint 2
TS-02	Develop the ML model for personalized dish recommendations so that the suggestions are accurate and relevant to user preferences. so that the suggestions are accurate and relevant to user preferences.	High	- Recommendation engine uses customer data. - Personalized suggestions are ranked accurately.	Sprint 2
TS-03	Set up a data pipeline to collect customer preferences and feedback so that the recommendation model stays up-to-date. so that the recommendation model stays up-to-date.	High	- Customer data is stored and pre-processed. - Data is fed into the ML model for training.	Sprint 2
US-09	As a customer, I want to see the price of each dish updated in real-time so that I can make informed purchase decisions. so that I can make informed purchase decisions.	High	- Prices adjust dynamically in real-time. - Users see updated prices on the menu.	Sprint 3

Product Backlog

ID	Story	Priority	Acceptance Criteria	Sprint
TS-04	Create backend infrastructure for real-time dynamic pricing updates so that the system can adjust prices based on demand. so that the system can adjust prices 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 3
US-10	As a customer, I want to remove items from my cart so that I can modify my order before placing it. so that I can modify my order before placing it.	Medium	<ul style="list-style-type: none">- Customers can remove items.- Cart updates immediately with accurate pricing.	Sprint 3
US-11	As a customer, I want to view a summary of my cart before checkout so that I can confirm order details. so that I can confirm order details.	Medium	<ul style="list-style-type: none">- Cart displays selected items and total cost.- Cart updates dynamically with pricing.	Sprint 3
US-12	As a customer, I want to place an order for recommended dishes so that I can complete my purchase conveniently. so that I can complete my purchase conveniently.	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

Product Backlog

ID	Story	Priority	Acceptance Criteria	Sprint
TS-05	Develop the dynamic pricing algorithm that adjusts prices in real-time so that pricing remains fair and competitive.	High	<ul style="list-style-type: none">- Pricing algorithm adjusts in real-time.- Customers see price changes in real time.	Sprint 3

Sprint 2 (Carried Forward) Backlog

ID	Story	Story Points	Acceptance Criteria
US-08	<p>User Story: Add Items to Cart</p> <p>As a customer, I want to add items to my cart so that I can easily manage my order. so that I can easily manage my order.</p>	5	<ul style="list-style-type: none">1. Customers can add items to their cart.2. Cart updates in real-time with selected dishes.
TS-04	<p>Technical Task: Real-Time Pricing Backend</p> <p>Create backend infrastructure for real-time dynamic pricing updates so that the system can adjust prices based on demand. so that the system can adjust prices based on demand.</p>	5	<ul style="list-style-type: none">1. Backend handles real-time price updates.2. Pricing changes are served to the frontend API.

Sprint 3 Backlog

ID	Story	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 so that I can make informed purchase decisions. so that I can make informed purchase decisions.	5	- 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 items from my cart so that I can modify my order before placing it. so that I can modify my order before placing it.	3	- 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 so that I can confirm order details. so that I can confirm order details.	2	- 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 recommended dishes so that I can complete my purchase conveniently. so that I can complete my purchase conveniently.	5	- Customers can place an order after adding items to the cart. - Successful order redirects to confirmation.

Sprint 3 Backlog

ID	Story	Story Points	Acceptance Criteria
TS-05	Technical Task: Dynamic Pricing Algorithm Develop the dynamic pricing algorithm that adjusts prices in real-time so that pricing remains fair and competitive.	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 that I can easily manage my order. so that I can easily manage my order.	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 so that the system can adjust prices based on demand. so that the system can adjust prices based on demand.	5	<ul 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	US-08: Add Items to Cart	Verify a user can add items to the cart from the personalized menu.	Items are successfully added to the cart and displayed in real-time.	Completed as expected
TC-02	US-08: Add Items to Cart	Verify the cart updates in real-time when items are added.	The cart dynamically updates with the newly added items immediately.	Completed as expected
TC-03	US-09: Dynamic Pricing Display	Verify prices of dishes adjust dynamically based on backend pricing changes.	Menu prices update in real-time to reflect current demand-based pricing.	Completed as expected
TC-04	US-09: Dynamic Pricing Display	Verify users can see updated prices without refreshing the page.	Prices change seamlessly in real-time on the frontend.	Completed as expected

Test Cases

ID	User Story / Task	Test Case	Expected Result	Actual Result
TC-05	US-10: Remove Items from Cart	Verify a user can remove items from the cart.	Items are successfully removed, and the cart updates immediately.	Completed as expected
TC-06	US-10: Remove Items from Cart	Verify modifying item quantities updates the cart correctly.	Quantity updates reflect correctly in the cart with real-time pricing adjustments.	Completed as expected
TC-07	US-11: View Cart Summary	Verify the cart displays all selected items and their prices accurately.	Cart shows all selected items, their individual prices, and the total cost dynamically.	Completed as expected
TC-08	US-11: View Cart Summary	Verify the total price updates dynamically if prices or quantities change.	The total price in the cart updates immediately based on pricing or quantity changes.	Completed as expected
TC-09	US-12: Place Order	Verify users can successfully place an order from the cart.	The order is processed successfully, and the user is redirected to the order confirmation page.	Completed as expected

Test Cases

ID	User Story / Task	Test Case	Expected Result	Actual Result
TC-10	US-12: Place Order	Verify the cart is cleared after order placement.	Cart becomes empty, and confirmation of the order is displayed to the user.	Completed as expected
TC-11	TS-04: Real-Time Pricing Backend	Verify the backend updates prices based on demand in real-time.	Pricing updates are reflected immediately in the frontend through the API.	Completed as expected
TC-12	TS-04: Real-Time Pricing Backend	Verify pricing changes do not result in application crashes or delays.	The system handles pricing updates efficiently without downtime.	Completed as expected
TC-13	TS-05: Dynamic Pricing Algorithm	Verify the algorithm adjusts prices correctly based on input parameters.	The pricing algorithm adjusts accurately according to demand and availability conditions.	Completed as expected
TC-14	TS-05: Dynamic Pricing Algorithm	Verify the algorithm handles edge cases, e.g., no demand or high demand.	Pricing remains consistent and logical under extreme scenarios.	Completed as expected

Sprint 3 Completed

ID	User Story / Task	Status
US-08	Add Items to Cart	Completed
US-09	Dynamic Pricing Display	Completed
US-10	Remove Items from Cart	Completed
US-11	View Cart Summary	Completed
US-12	Place Order	Completed
TS-04	Real-Time Pricing Backend	Completed
TS-05	Dynamic Pricing Algorithm	Completed

Metrics

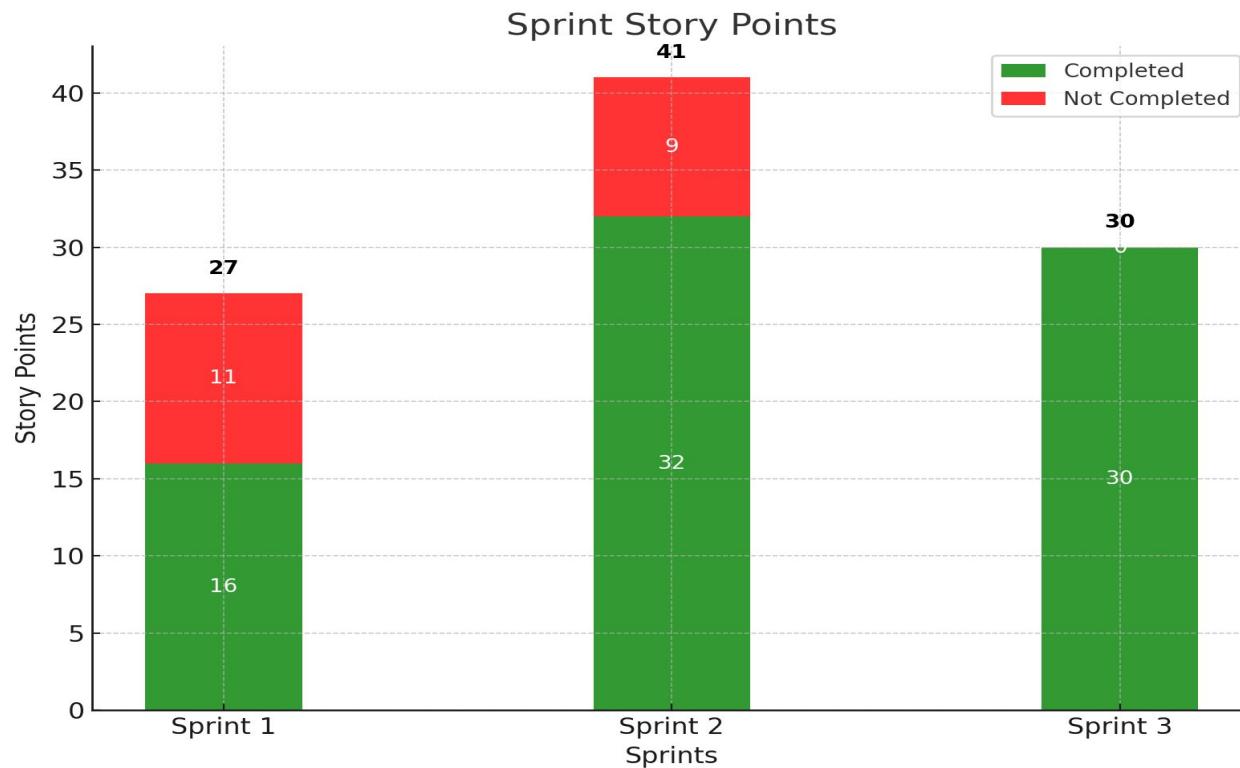
Team Velocity - Sprint 3

- **Total Story Points Committed:** 30 story points
- **Total Story Points Completed:** 30 story points

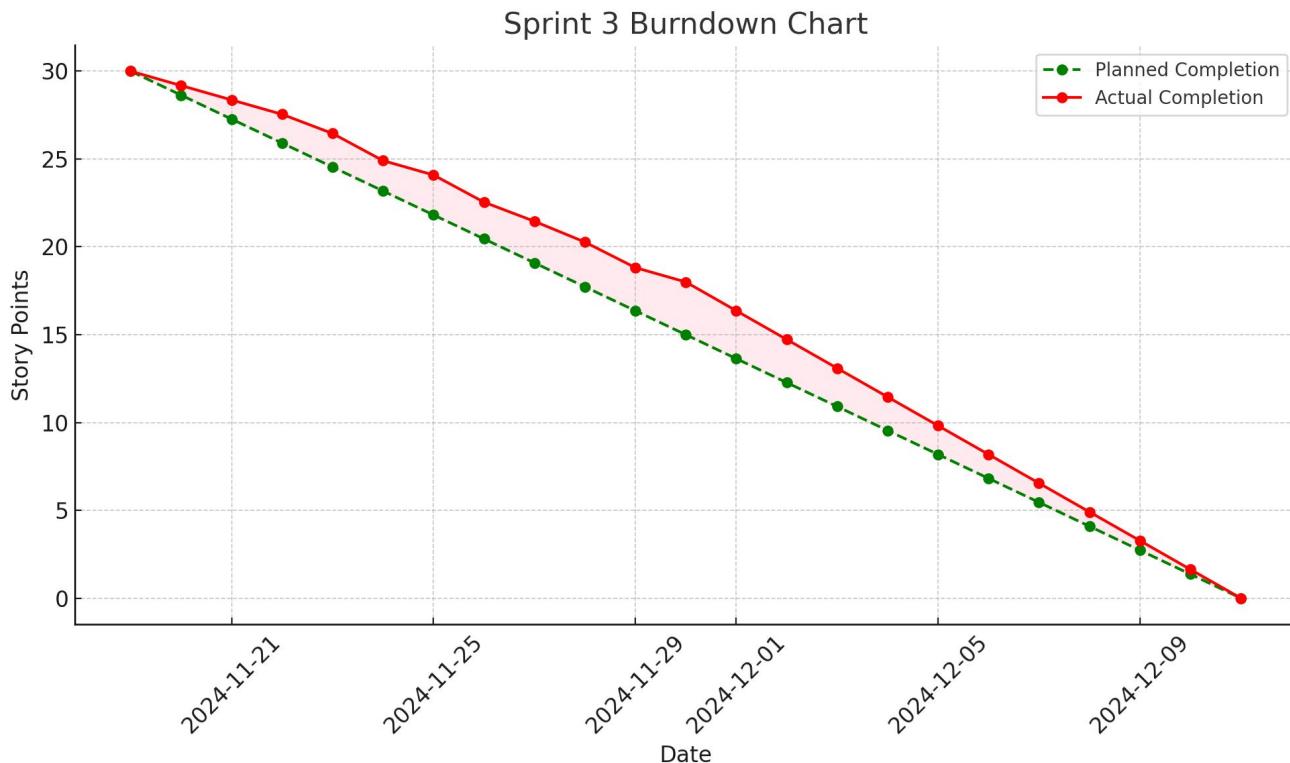
Completed/Committed Ratio

- **Committed Story Points:** 30 story points
- **Completed Story Points:** 30 story points
- **Thus, the Completed/Committed Ratio for Sprint 3 is 100.00%.**

Historical Velocity for Sprints



Burndown Chart



Retrospective

[Export](#)[Login](#)

SPRINT 3

What went well

Tools and technology enhanced productivity.	Good team collaboration and efficient communication with each other.
+2	+2
Peer programming and code reviews enhanced learning and reduced errors.	Daily stand ups helped in clarity on tasks and their progress.
+4	+4

What can be improved

Reflect on how feedback was addressed and what could be done differently.	External dependencies delayed some tasks; better coordination is needed.
+4	+2
Less time allocated to test cases, which lead to last minute fixing.	Tasks took longer than estimated, indicating room for better sprint planning.
+1	+1

Action Items

Schedule regular grooming sessions to clarify and refine requirements.	Include visual aids to improve understanding of user stories.
+4	+1
Use past sprint data to better estimate the time required for similar tasks.	Follow and abide to the schedules.
+2	+3

Upskill team members to reduce dependence on external resources when possible. Upskill team members to reduce dependence on external resource.
+4

As we already have our personal deadlines, just we need to stick to them and update to the team.
+2

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

Deactivate

Website Screenshots

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

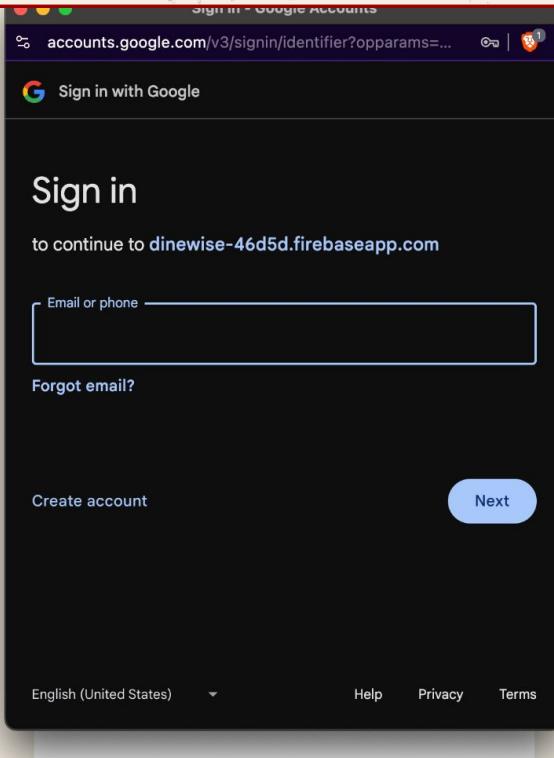
The main dashboard features several key metrics: "Total Revenue" (\$0.00), "Avg. Order Value" (\$0.00), and a search bar labeled "Search menu items...". Below these, the "Menu Management" section includes a search input and a table listing menu items. One item, "Crispy Spring Rolls", is highlighted with a red bar at the bottom.

A central modal window titled "Add New Menu Item" is open, prompting the user to enter details for a new item. The fields include:

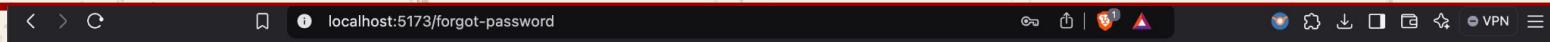
- Name:** An input field where the user has typed "Name".
- Description:** A large text area for the item description, currently empty.
- Price:** An input field containing a dollar sign (\$) and a dropdown menu labeled "Select Category".
- Preparation Time (minutes):** An input field for preparation time, currently empty.
- Dietary Restrictions:** A section with two columns of checkboxes:
 - Left column: Vegetarian, Gluten-Free, Nut-Free, Kosher.
 - Right column: Vegan, Dairy-Free, Halal.

At the bottom of the modal are "Cancel" and "Save Item" buttons.

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...". 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 shown: "bhavya" (Various Cuisine, 2728 john f kennady ,blvd, 15714360352), "deekshitha" (Various Cuisine, 2728 john f kennady ,blvd, 15714360352), and "phani" (Various Cuisine, 215 linden ave, 83748928787). Each card has a "View Menu >" button at the bottom.

DineWise

Welcome back!

Discover new restaurants and explore your favorite cuisines

All Indian Italian Chinese Mexican Japanese American

Search restaurants...

bhavya ★
Various Cuisine
④ 2728 john f kennady ,blvd
⌚ 15714360352

deekshitha ★
Various Cuisine
④ 2728 john f kennady ,blvd
⌚ 15714360352

phani ★
Various Cuisine
④ 215 linden ave
⌚ 83748928787

[View Menu >](#)

[View Menu >](#)

[View Menu >](#)

Update Preferences

Website Screenshots

[← Back to Restaurants](#)

Spice Garden

4.5

Indian

123 Curry Lane, Foodville

(555) 123-4567

Current time: 2:51 PM

Prices will update in: 9 minutes

Menu

Butter Chicken 🌶

Creamy tomato-based curry with tender chicken

Gluten-Free

\$18.47

\$15.99

+ Add to Cart

Palak Paneer

Spinach curry with cottage cheese

Vegetarian Gluten-Free

\$16.16

\$13.99



Website Screenshots

Prices will update in: 30 minutes

Menu

Palak Paneer Recommended

Spinach curry with cottage cheese

Vegetarian Gluten-Free

+ Add to Cart

Dal Makhani Recommended

Creamy black lentils simmered overnight

Vegetarian Gluten-Free

+ Add to Cart

Malai Kofta Recommended

Cheese and vegetable dumplings in rich cream sauce

Vegetarian

+ Add to Cart



Website Screenshots

[← Back to Restaurants](#)

Spice Garden

Indian

📍 123 Curry Lane, Foodville
🕒 (555) 123-4567
🕒 Current time: 2:51 PM
Prices will update in: 9 minutes

Menu

Butter Chicken ⚡
Creamy tomato-based curry with tender chicken
Gluten-Free
[+ Add to Cart](#)

Palak Paneer
Spinach curry with cottage cheese
Vegetarian Gluten-Free

Your Order X

Palak Paneer \$16.17 each	<input type="button" value="-"/> <input type="button" value="1"/> <input type="button" value="+"/> <input type="button" value="X"/>
Vegetable Samosas \$8.08 each	<input type="button" value="-"/> <input type="button" value="1"/> <input type="button" value="+"/> <input type="button" value="X"/>
Lamb Rogan Josh \$21.94 each	<input type="button" value="-"/> <input type="button" value="1"/> <input type="button" value="+"/> <input type="button" value="X"/>

Total: **\$46.19**

[Proceed to Checkout](#)

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 options. 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:

```
1 [ "recommendations": [ 2 { 3 "category": "Italian", 4 "description": "Fresh vegetables with gluten-free pasta", 5 "features": "Pasta Primavera Fresh vegetables with gluten-free pasta Italian vegetarian", 6 "id": "1", 7 "is_spicy": false, 8 "is_vegetarian": true, 9 "name": "Pasta Primavera" }, 10 { 11 "category": "Italian", 12 "description": "Classic pizza with gluten-free crust and fresh basil", 13 "features": "Margherita Pizza Classic pizza with gluten-free crust and fresh basil Italian vegetarian", 14 "id": "3", 15 "is_spicy": false, 16 "is_vegetarian": true, 17 "name": "Margherita Pizza" }, 18 { 19 "category": "Indian", 20 "description": "Mild curry with seasonal vegetables", 21 "features": "Vegetable Curry Mild curry with seasonal vegetables Indian vegetarian", 22 "id": "5", 23 "is_spicy": false, 24 "is_vegetarian": true, 25 "name": "Vegetable Curry" } ] }
```

The response status is 200 OK, with a time of 32 ms and a size of 1.71 KB. Below the response, there are tabs for Body, Cookies, Headers, Test Results, and a preview of the JSON data. The bottom of the screen features a toolbar with icons for Online, Find and replace, Console, and various system functions like 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