# Project “*Pour Rice*” Proposal

Cross Platform Assignment — Pour Rice / 倒米

Author: Ng Yu Ham Baldwin

## 

## 1. Project overview

This project entails the development of a cross-platform food discovery application, inspired by the structure of OpenRice. The application will be constructed using Ionic with Angular and Capacitor to streamline compatibility with Progressive Web Application (PWA) standards. Furthermore, it will adopt a mobile-first design approach to demonstrate its capabilities in my portfolio effectively.

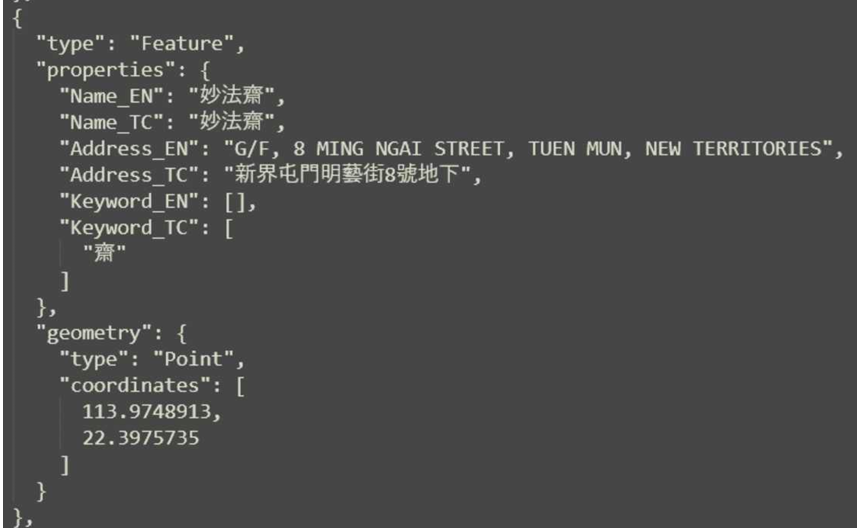
### 2. Motivation and aims

The primary motivation of this project is to demonstrate the development of a modern, accessible, and bilingual cross-platform application prototype that aims:  
- Create a visually appealing and responsive food discovery prototype for both mobile and desktop platforms;  
- Demonstrates modern full‑stack skills (Ionic/Angular, Tailwind CSS, Node.js, Firebase);  
- Implements bilingual UI (EN / TC) and a light/dark theme toggle for accessibility and polish;  
- Exposes searchable restaurant data and user interactions for portfolio presentation.  
Secondary aims include:   
- Saves multiple menus as a document nested in the Restaurant with BLOB to store 1-4MB images and dish details.  
- Invite the community/users to add unassigned restaurants along with their menus and dishes to showcase options that do not have claimed owners.   
- Allowing users to see available seats/tables via a calendar of a restaurant, then book at a specified time.

### 3. Objectives and deliverables

Deliver a working demo containing:  
- Home page: Includes advertisement carousel, featured offers, articles, latest reviews and trending restaurants;  
- Search page: Allows keyword-based search of restaurants (mocked data) and tags (veg types and districts);  
- Restaurant page: Displays menus, locations, opening hours, reviews and other restaurant details;  
- User page: Editing profile;  
- Theme and Language Services: Fully functional toggles for dark/light themes and English/Chinese language.

### 4. Technical stack

- Framework: Ionic + Angular (single codebase for web and mobile).  
- Styling: Tailwind CSS v4 for utility-first styling, responsive layouts, and dark mode support.  
- Interaction: Swiper for touch‑friendly carousels (ads, menus).  
- Data: Firebase Authentication and Firestore for rapid prototyping and hosting;  
 The database documents will include: Restaurant(GeoJSON), User(create by Firebase OAuth), Menu  
  
  
  
   
- Development: Node.js for local tooling;

### 5. Implementation plan & milestones

Week 1-2 — Project skeleton and UI: initialise Ionic/Angular app, integrate Tailwind, implement theme and language services, create page scaffolding (home, search, restaurant, user) and mock JSON data.  
Week 3-4 — Interactive components: integrate Swiper for carousels, implement responsive card feeds, and wire up language and theme persistence. Then make those modules.  
Week 5-6 — Data & authentication: integrate Firebase Auth (OAuth) and Firestore (read/write sample data), implement restaurant detail and review submission flows.  
Week 7-8 — Accessibility: Create an Elderly mode on mobile end; Or Search: Integrate Algolia or emulate search with Firestore; perform accessibility checks, ensure responsiveness, implement minor performance optimisations, and prepare the deployment pipeline for Vercel.  
Week 9 — Testing & submission: Final testing, prepare presentation materials, and push the final repository.

### 6. Risks and mitigations

- Dependency and build issues (Ionic + Tailwind cross‑config): mitigate by locking Node LTS, using Swiper's Angular module, and isolating Tailwind build steps.  
- Time constraints: present a minimum demonstrable MVP (Home, Search, Restaurant, User) with mocked data; postpone advanced features (reservations, calendar) to future work.

### 7. Evaluation and testing

Evaluation will focus on these aspects:  
- Functionality: Correct operation of navigation, carousel, and toggles.  
- Responsiveness: Seamless scaling between mobile and web viewports.  
- Usability: Intuitive layout, straightforward typography, and accessible language switching.  
- Visual Consistency: Correct application of light/dark modes across all elements.  
Testing will be performed using browser developer tools and Android/iOS emulators.

### 8. Resources and prior work

- Adapt code and assets from last year’s repository (Web‑Assignment).  
Current progress includes:  
- Working language and theme services;  
- Home page UI with functional layout and test data;  
- Initial Swiper carousel implementation;

### 9. Expected outcomes

The minimum viable product aims to be a well-designed, cross-platform prototype that functions seamlessly as both a web application and a mobile application. This prototype will showcase key design principles, support multiple languages, offer theme customisation, and provide a responsive user interface. Additionally, it will feature basic search functionality and demonstrate interactions with a sample database, ultimately setting a solid foundation for future development.  
The final repository will be organised, documented and suitable for portfolio inclusion next year. Repo:<https://github.com/Test-Plus-XD/Cross-Platform-Assignment.git>

### 10. Possible forecast

- On the Restaurant page, save reservation data. This could be a bulk patch to remove outdated data using Firestore rules or script runs on Render.  
- On the Restaurant page, add Gemini. When a user asks a question, fetch the database for that restaurant and use a text parser, then send it to AI. (NPM)  
- On the Mobile end, send a notification on the event trigger.  
- Let the restaurant owner display advertisements, then integrate Stripe for fee collection.  
- Use Algolia to replace search.  
- Use Make to send an email to the notification on the reservation.  
- Buy a 7USD domain on Cloudflare, then build on Vercel in the far future.