

## Feasibility

This document will outline the feasibility of creating the **Restaurant** application.

## Usecase Overview

**Users** should have the ability to book a table at the restaurant at an available time; users should also have the ability to view the restaurant menu, create an account, log in and order food online through the restaurant application.

The restaurant staff (**admins**) should have the ability to view both food orders and bookings that users placed. Admins should also have the ability to add, delete and edit menu items through the restaurant application.

## Risks/Considerations

- There is a great deal of work required in a fairly strict timescale; due to this time pressure, it might not be feasible to complete each requirement to its full extend and hence there might be requirements that have room for improvement. Therefore, prioritising the core/essential requirements to deliver an MVP (minimum viable product) is essential.
- Users can sign in and log out; this should be done securely to keep their data safe; this is also a legal requirement for personal information to be handled securely - this means only the user and the restaurant should have access to that data.
- Users can submit personally identifiable information using the booking feature because sensitive data is handled; this should also be stored and accessed securely and hence should not be accessible publically or by any unauthorised person
- The platform needs to be easy to use, as users of many ages and technical know-how might use it
  - Therefore, the restaurant application should use a modern, intuitive UI
  - It should work out of the box/ have minimal setup required
  - It should be performant so that it works on slower devices that might be used by some individuals

## Technology Overview

Domain/Feature	Technology
Frontend	Nuxt.js ( <i>Vue.js</i> )
Backend	Express.js
Testing	Jest
Database	Firestore
Deployment/ Cloud Provider	Google Cloud Platform
Source Control	GitHub

## Decision reasons

**Vue/Nuxt** Vue is an easy to use progressive framework for building modern user interfaces; nuxt is built on top of Vue and provides additional features such as: - Pre-configured web framework that automatically serves the client the Vue application - SSR (Server-Side Rendering) - File-system routing for asset management - Smart code splitting for enhanced performance - Easy to use and configure Router for page routing/navigation - And Much More

This will make it easier to build a modern performant application. The Nuxt application also supports Progressive Web Apps (PWA), which means it can be modified to be a fully-fledged mobile app and as well a desktop application if that was ever a requirement in the future.

**Express** Express is an industry-standard web framework more than capable of creating a restful API that can be used in this application

**Firestore (NoSQL)** Firestore is a cloud-hosted NoSQL database; it allows for greater flexibility than a typical relational database by having a flexible schema, which means data can be semi-structured or even completely unstructured. This provides much room for future development; for example, the menu item collection (similar to a SQL table) can have simple menu items for now, but in the future, a new item can have unique special properties that do not need to be added to all the other items. Other than the added flexibility and increased development velocity, A NoSQL database is also available in the free tier of Google Cloud Platform under the service called **Firestore**, unlike Google's SQL databases, which would incur a charge to run on the Google Cloud Platform. Firestore also allows you to run sophisticated ACID transactions to keep data integrity. Lastly, Firestore also has a feature that enables realtime updates; this can be especially useful for the menu, which has a limited stock amount; every time someone makes an order, any other user looking at the menu will see it in realtime if an item goes out of stock.

**Jest** Jest is also an industry standard Javascript Testing framework, and because Vue/Nuxt use Javascript, staying within the same programming language will decrease context/language switching, which will help will development velocity

**Google Cloud Platform** GCP is an amazing cloud provider that is easy to use, and for the purpose of this application, it will fall within its free tier. GCP has services that I will need to use for the application; these tools include: - **App Engine/Cloud Run** - Services that can be used to deploy the app and have it accessible publically - **Cloud Functions** - Backend serverless functions triggered by firebase events and HTTP requests - **Firestore** - A NoSQL database to store User, Order & Booking data - **Cloud Build** - A service to auto build and deploy the app based on events such as a git pushes to the repo

**GitHub** Github is free to use source control software that provides many features and is one of the industry standards for source control, and I have experience using it.