

Quiz Application Using Front-End Framework

This document contains sections for:

- **Sprint planning and Task completion.**
- **Core Concepts used in the project.**
- **Technologies and Tools Used in the Project.**
- **The flow of the Application.**
- **Demonstrating the product capabilities, appearance, and user interactions.**
- **Unique Selling Points of the Application.**
- **Conclusions.**

- The code for this project is hosted at https://github.com/Dinesh123527/Simpli-phase_4_project_quizando
- This project is developed by **G V Narasimha Raju.**

Sprints planning and Task completion:

The project is planned to be completed in 2 sprints. Tasks assumed to be completed in the sprint are:

1st Sprint:

- Creating the flow of the application.
- Initializing the git repository to track changes as development progresses.
- Creating an Angular application to fulfill user requirements.
- Adding the Required Packages used for the application.
- Creating Components and Services, Styling the Pages.

2nd Sprint:

- Implemented the JSON Part by Initializing the Server.
- Creating Reactive Forms and Validations to the Forms.
- Creating Service methods to fetch the response.
- Passing the Response from the Service to the component class.
- Created the Quiz data and mapped the data accordingly.
- Implement the Shuffling Logic for each time Question loading i.e., getting random questions every time.

Core concepts used in the project:

- **Object-Oriented:** used to create and model objects for users and their credentials.
- **Reactive Forms:** Reactive forms provide a model-driven approach to handling form inputs whose values change over time along with Validations.
- **Component:** Angular components are a subset of directives, always associated with a template.
- **Lifecycle hooks:** `ngOnInit()`, `ngAfterViewInit()`
- **Data Binding:** Data binding deals with how to bind your data from component to HTML DOM elements (Templates). We can easily interact with applications without worrying about how to insert our data. We can make connections in two different ways “1-way” and “2-way binding”.
- **Directives:** In Angular, Directives are defined as classes that can add new behavior to the elements in the template or modify existing behavior.
- **Routing:** The process of defining the navigation element and its associated view is called routing in Angular. Angular provides a separate module, the Router module, for setting up navigation in an Angular application.
- **Services:** Angular Services is a piece of reusable code with a focused purpose. A code that you will use across multiple components in your application.
- **RXJS Concepts:** RxJS provides an implementation of the Observable type, which is needed until the type becomes part of the language. The library also provides utility functions for creating and working with observables.

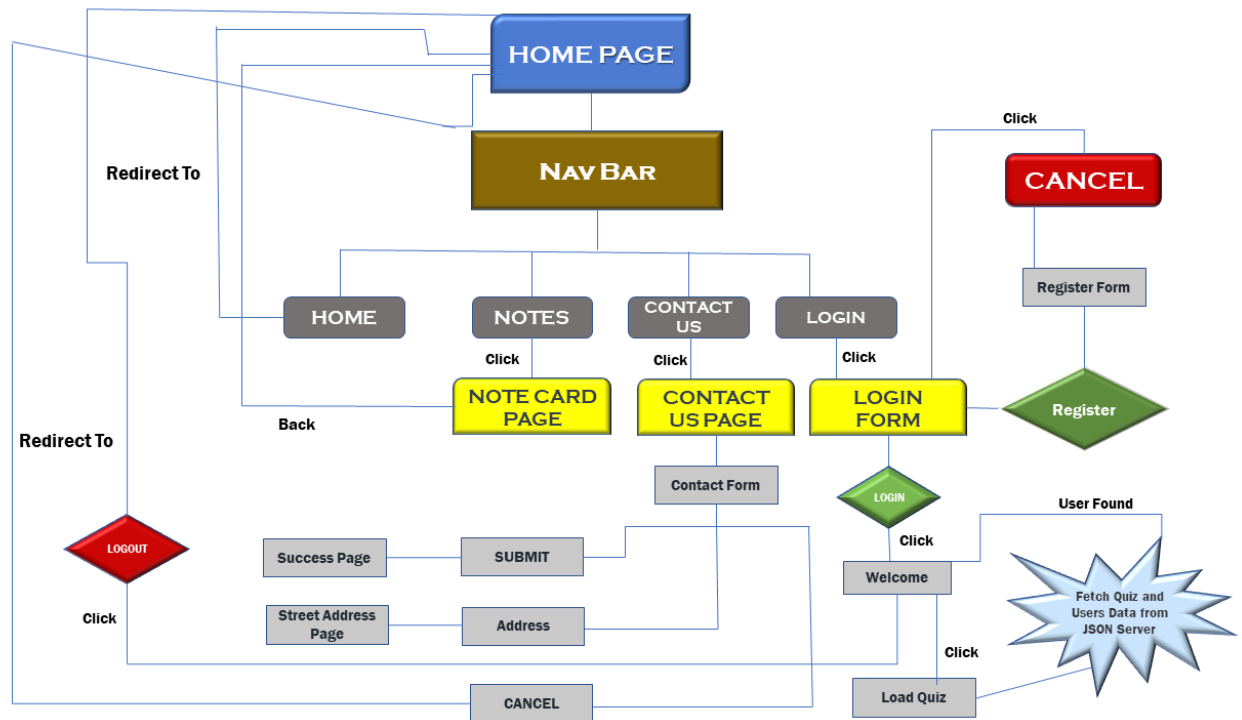
Subscribe, Observer, Subject, Behavior Subject.

- **JSON Server:** JSON Server is a Node Module that you can use to create rest JSON web service.
- **API:** API stands for application programming interface, which is a set of definitions and protocols for building and integrating application software.

Technologies and Tools Used in the Project

- **Angular:** To create Quiz Application.
- **Angular Material:** Angular Material is a User Interface (UI) component library that developers can use in their Angular projects to speed up the development of elegant and consistent user interfaces.
- **JSON Server:** JSON Server is a Node Module that you can use to create the rest JSON Web Service.
- **Visual Studio Code:** Visual Studio Code is a streamlined code editor with support for development operations like debugging, task running, and version control. It aims to provide just the tools a developer needs for a quick code-build-debug cycle.

Flow of the Application:



Demonstrating the product capabilities, appearance, and user interactions:

To demonstrate the product capabilities, below are the sub-sections configured to highlight appearance and user interactions for the project.

- ✓ [Creating new Angular project using ng new project name.](#)
- ✓ [Add the required packages to the project via node package manager npm command.](#)
- ✓ [Open the Angular Project in that go to package.json to view the added dependencies.](#)
- ✓ [Now Run the code using ng serve command and the application will be deployed in the default port of localhost:4200.](#)

- ✓ **Pushing the code to the GitHub repository.**

Unique Selling Points of the Application:

- ✓ This was Quiz Application developed using Angular Frame work where user can login using data from JSON Server and User can also be Registered.
- ✓ The Application has only one Functionality i.e., a User can only Login to the Portal and take the Quiz.
- ✓ Getting the Quiz Data i.e., Question for the Quiz Randomly on every time and having the Timer for the Quiz with nice User Interface Design and Provided Rich UI Design Overall the Application.

Conclusions:

Further enhancements to the application can be made which may include:

- Separating the roles for User's and Admin Part such that the admin Can manage the website such as adding the questions and maintain Quiz data dynamically.
- Need to Validate the User's dynamically i.e., by adding some authentication method and need to Implement Authenticated Routing.
- Adding the Different Quiz's let user select the quiz of his wish and populate the Quiz data respectively.

