|  |
| --- |
|  |
| KITCHEN-STORY |
| Created by Shaan Ngwenzane |

­­

Problem Statement:

Description:

The Kitchen Story Application system creates an application that enables users to buy kitchen items like vegetables, leafy vegetables, fruits and dry fruits. It provides an easy way to purchase the items in flexible hours of user. Admin can authorize the users and can manage the kitchen items.

System Details:

This system contains four main module:Kitchen items,Item Details,Cart and Admin.The user opens the kitchen items module and search for desired item. The user can directly buy if user want only one item else the user can see the item details and add to the cart.The user can modify the quantity details in the cart module and finally checkout and place an order.Now admin can authorize the users while users logging in and admin can change his password .Admin can view the item details, add and delete if necessary.

Angular:

Angular is an application design framework and development platform for creating efficient and sophisticated single-page apps.

Angular is a development platform, built on [TypeScript](https://www.typescriptlang.org/). As a platform, Angular includes:

* A component-based framework for building scalable web applications
* A collection of well-integrated libraries that cover a wide variety of features, including routing, forms management, client-server communication, and more
* A suite of developer tools to help you develop, build, test, and update your code

With Angular, you're taking advantage of a platform that can scale from single-developer projects to enterprise-level applications. Angular is designed to make updating as easy as possible, so you can take advantage of the latest developments with a minimum of effort. Best of all, the Angular ecosystem consists of a diverse group of over 1.7 million developers, library authors, and content creators.

Angular CLI:

The Angular CLI is a command-line interface tool that you use to initialize, develop, scaffold, and maintain Angular applications directly from a command shell.

npm install -g @angular/cli

Components:

Components are the building blocks that compose an application. A component includes a TypeScript class with a @[Component](https://angular.io/api/core/Component)() decorator, an HTML template, and styles. The @[Component](https://angular.io/api/core/Component)() decorator specifies the following Angular-specific information:

* A CSS selector that defines how the component is used in a template. HTML elements in your template that match this selector become instances of the component.
* An HTML template that instructs Angular how to render the component.
* An optional set of CSS styles that define the appearance of the template's HTML elements.

The following is a minimal Angular component.

import { Component } from '@angular/core';

@Component({

selector: 'hello-world',

template: `

<h2>Hello World</h2>

<p>This is my first component!</p>

`,

})

export class HelloWorldComponent {

// The code in this class drives the component's behavior.

}

To use this component, you write the following in a template:

<hello-world></hello-world>

When Angular renders this component, the resulting DOM looks like this:

<hello-world> <h2>Hello World</h2> <p>This is my first component!</p></hello-world>

Templates:

Every component has an HTML template that declares how that component renders. You define this template either inline or by file path.

Angular extends HTML with additional syntax that lets you insert dynamic values from your component. Angular automatically updates the rendered DOM when your component’s state changes. One application of this feature is inserting dynamic text, as shown in the following example.

<p>{{ message }}</p>

The value for message comes from the component class:

import { Component } from '@angular/core';

@Component ({

selector: 'hello-world-interpolation',

templateUrl: './hello-world-interpolation.component.html'

})

export class HelloWorldInterpolationComponent {

message = 'Hello, World!';

}

View Components:

1.Home Component

2.Aboutus Component

3.Contactus Component

4.Register Component

5.Login Component

6.Kitchen Item Component

7.Kitchen Details Component

8.Cart Component

9.Payment Component

10.Success Component

11.Admin Component

12.Change Password Component

13.View details Component

14.Add Item Component

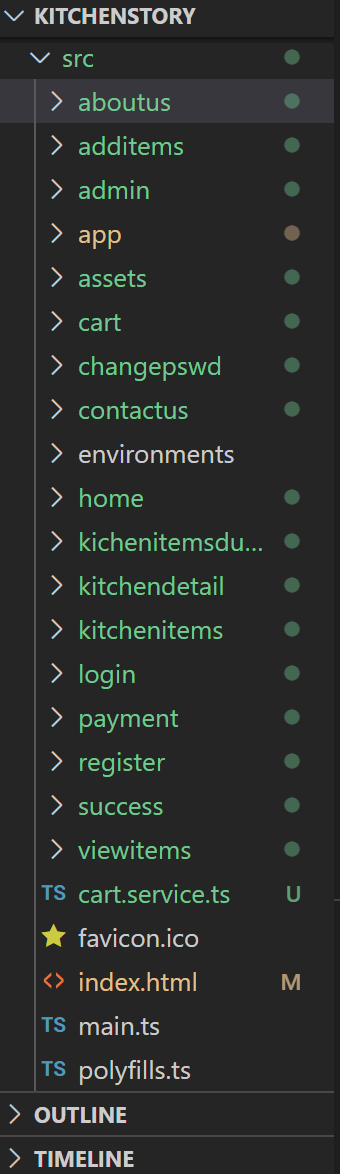


Fig: Files of visual studio code