Ionic

- Open source UI toolkit that includes over 100 components
- Works with Angular, React, Vue and other frameworks
- Cross-platform with adaptable user interface
- Platform for creating hybrid mobile apps

Key features

- Cross-platform
- Integration with Angular
- Component Library
- Capacitor
- Performance
- Development Efficiency
- Community and Support

Native

- Proficiency in each platform required
- Entirely separate code bases
- Timely & expensive development

Hybrid

- HTML5 that acts like native
- Web wrapped in the native layer
- Direct access to native API's

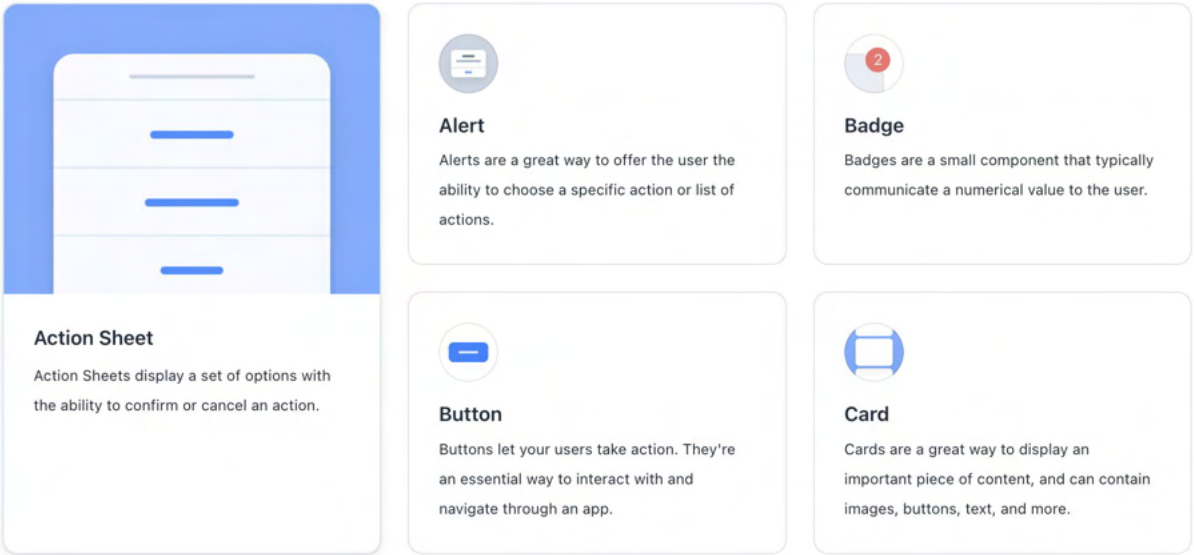
Ionic CLI

- To install the CLI: **npm install -g @ionic/cli**
- To create a project: **ionic start projectname template**
- To create an angular/ionic app: **ionic start projectname tabs --type=angular**
- To test in a browser: **ionic serve**
- To see all templates available: **ionic start --list**

Install & Configure

- To create a new ionic project: **yourprojectname tabs --type=angular / ionic start**
- To create a new project: **ionic serve -o**
- Generate something new: **ionic generate**

Ionic components



Ionic

Service API integration

- A service is a way to share data, logic, or functions across different parts of your application.
- Services are classes that are used to encapsulate and manage specific functionalities, making it easier to maintain and test your code.

ionic g s yourservicename

Code example

```
import { Injectable } from '@angular/core';

@Injectable({
   providedIn: 'root', // This makes the service available application-wide
})

export class MyService {
   constructor() { }

   myMethod() {
       console.log('This is my service method!');
   }
}
```

hero.service

Code example

```
[HttpGet]
[Route("GetAllHeroes)]

public async Task<IActionResult> GetAllHeroes()
{
    try
    {
         var results = await _heroRepository.GetAllHeroesAsync();
         return Ok(results);
    } catch (Exception) {
         return StatusCode(500, "Internal Server Error.  Please contact support.");
    }
}

[HttpGet]
[Route("GetHero/{heroId}")]

public async Task<IActionResult> GetHero(int heroId)
{
    try
    {
         var result = await _heroRepository.GetHeroAsync(heroId);

         if (result == null) return NotFound("Hero does not exist.  You need to
         create it first");

         return Ok(result);
    } catch (Exception) {
         return StatusCode(500, "Internal Server Error.  Please contact support");
    }
}
```

API controller

Ionic

Routing

Routing in Angular/Ionic is the process of defining paths in your application and linking those paths to specific components. This allows users to navigate between different views or pages within your app. Let's break it down step by step and include a simple code example.

Step-by-Step Explanation

1. Setup Routes: In Angular/Ionic, routes are defined in a module's routing file. Each route specifies a path and the component that should be displayed when the path is accessed.
2. Router Module: The Angular Router is used to navigate between views or components. You need to import the RouterModule and configure it with your routes.
3. Navigation: Links (<a>) or buttons can be used to navigate to different routes using the routerLink directive.

```
import { NgModule } from '@angular/core';
import { PreloadAllModules, RouterModule, Routes } from '@angular/router';
import { HomeComponent } from './pages/home/home.component';
import { AboutComponent } from './pages/about/about.component';

const routes: Routes = [
  { path: '', redirectTo: 'home', pathMatch: 'full' }, // Redirect to home by default
  { path: 'home', component: HomeComponent },
  { path: 'about', component: AboutComponent },
];

@NgModule({
  imports: [
    RouterModule.forRoot(routes, { preloadingStrategy: PreloadAllModules })
  ],
  exports: [RouterModule]
})
export class AppRoutingModule { }
```

Code example ● app-routing.module.ts

```
<ion-app>
  <ion-header>
    <ion-toolbar>
      <ion-title>Routing Example</ion-title>
    </ion-toolbar>
  </ion-header>
  <ion-content>
    <ion-button routerLink="/home">Home</ion-button>
    <ion-button routerLink="/about">About</ion-button>
    <router-outlet></router-outlet>
  </ion-content>
</ion-app>
```

component.html

Ionic

Practical code

Backend ● Controller

```
import { Injectable } from '@angular/core';
import { HttpClient, HttpHeaders } from '@angular/common/http';
import { map, Observable, Subject } from 'rxjs';
import { Hero } from '../shared/hero';

@Injectable({
  providedIn: 'root'
})
export class HeroService {

  apiUrl = 'http://localhost:5116/api/'

  httpOptions ={
    headers: new HttpHeaders({
      ContentType: 'application/json'
    })
  }

  constructor(private _httpClient: HttpClient) { }

  getHeroes(): Observable<Hero[]>{
    return this._httpClient.get<Hero[]>(`${this.apiUrl}Hero/GetAllHeroes`)
    .pipe(map(result => result))
  }

  getHero(heroId: number) {
    return this._httpClient.get(`${this.apiUrl}Hero/GetHero` + "/" + heroId)
    .pipe(map(result => result))
  }
}
```

Frontend ● Shared model

```
export interface Hero {
    heroId: Number;
    name:String;
    age:number;
    birthday:String;
    height:string;
    isAlive:boolean;
    filename:String;
    imageBase64:String;
}
```

Ionic

Practical code

```
using Microsoft.AspNetCore.Components.Forms;
using Microsoft.AspNetCore.Http;
using Microsoft.AspNetCore.Mvc;
using System.Reflection.Metadata.Ecma335;

namespace Ionic_II.Controllers
{
    [Route("api/[controller]")]
    [ApiController]
    public class HeroController : ControllerBase
    {
        private readonly IHeroRepository _heroRepository;

        public HeroController(IHeroRepository HeroRepository)
        {
            _heroRepository = HeroRepository;
        }

        [HttpGet]
        [Route("GetAllHeroes")]
        public async Task<IActionResult> GetAllHeroes()
        {
            try
            {
                var results = await _heroRepository.GetAllHeroesAsync();
                return Ok(results);
            }
            catch (Exception)
            {
                return StatusCode(500,"Internal Server Error. Please contact support.");
            }
        }

        [HttpGet]
        [Route("GetHero/{heroId}")]
        public async Task<IActionResult> GetHero(int heroId)
        {
            try
            {
                var result = await _heroRepository.GetHeroAsync(heroId);

                if (result == null) return NotFound("Hero does not exist. You need to create it first");

                return Ok(result);
            }
            catch (Exception)
            {
                return StatusCode(500, "Internal Server Error. Please contact support");
            }
        }
    }
}
```

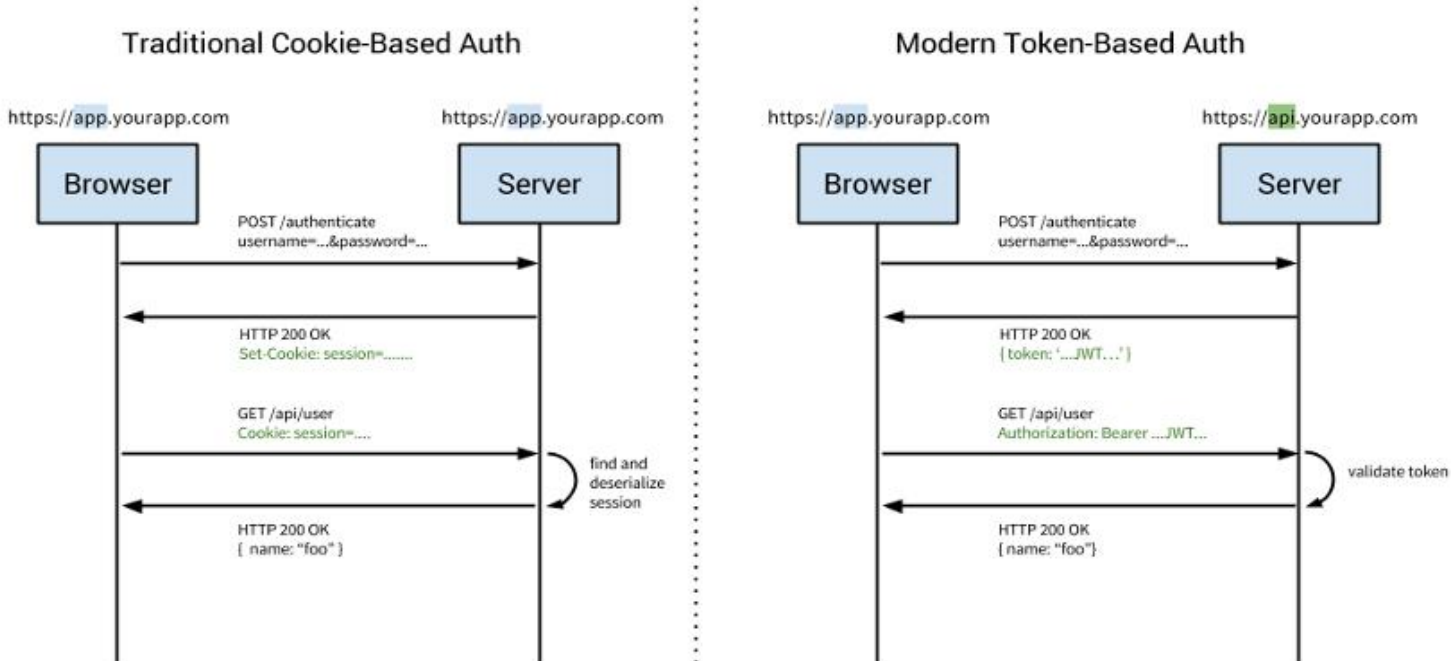
Application Security

- **Authentication**: Check to see if a person exists
- **Authorize**: Check what roles a user has and what can the user do
- **Audit**: Log what the user did
- Without any sort of authentication and authorization, anyone with the link to the API can access the data and get private company information, like customer details and it can cause a lot of problems
- This is not only to protect your own resources, but the confidentiality of your end users as well

Response codes

- **200 OK:** Everything successful and resources could be fetched
- **404 Not Found**: Server cannot find requested resources
- **403 Forbidden**: You don't have access rights to the content
- **401 Unauthorized**: You are not authenticated to request a resource
- **400 Bad Request**: Client issue - server cannot/will not process the request
- **500 Internal Server Error**: Something has gone wrong on the server

Authorization



Encoded

eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiIxMjM0NTY3ODkwIiwibmFtZSI6IkpvaG4gRG9lIiwiYWRtaW4iOnRydWV9.TJVA95OrM7E2cBab30RMHrHDcEfxjoYZgeFONFh7HgQ

Decoded

HEADER:

```
{
  "alg": "HS256",
  "typ": "JWT"
}
```

PAYLOAD:

```
{
  "sub": "1234567890",
  "name": "John Doe",
  "admin": true
}
```

VERIFY SIGNATURE

```
HMACSHA256(
  base64UrlEncode(header) + "." +
  base64UrlEncode(payload),
  secret
) ☐secret base64 encoded
```

Application Security

.NET Identity Installations

Microsoft.AspNetCore.Identity.EntityFrameworkCore by Microsoft — 5.0.17 / 6.0.5
ASP.NET Core Identity provider that uses Entity Framework Core.

Microsoft.Extensions.Identity.Core by Microsoft — 5.0.17 / 6.0.5
ASP.NET Core Identity is the membership system for building ASP.NET Core web applications, including membership, login, and user data. ASP.NET Core Identity allows you to add login features to your applic...

Microsoft.Extensions.Identity.Stores by Microsoft — 5.0.17 / 6.0.5
ASP.NET Core Identity is the membership system for building ASP.NET Core web applications, including membership, login, and user data. ASP.NET Core Identity allows you to add login features to your applic...

Microsoft.AspNetCore.Authentication.JwtBearer by Microsoft — 5.0.17 / 6.0.5
ASP.NET Core middleware that enables an application to receive an OpenID Connect bearer token.

Practical code

```
import { Component } from '@angular/core';
import { Router } from '@angular/router';
import { DataService } from '../services/data.service';

@Component({
  selector: 'app-login',
  templateUrl: './login.component.html',
  styleUrls: ['./login.component.scss'],
})
export class LoginComponent {
  courses: any[] = [];

  constructor(
    private router: Router,
    private dataService: DataService,
  ) {}

  Login() {
    this.dataService
      .Login()
      .subscribe((result: any) =>
        localStorage.setItem('Token', JSON.stringify(result)),
      );
  }

  GetCourses() {
    this.dataService.Courses().subscribe((result: any[]) => {
      this.courses = result;
    });
  }
}
```

Frontend ● login.rs

Application Security

Practical code

```
<div class="login-wrapper" fxLayout="row" fxLayoutAlign="center center">
  <button mat-stroked-button color="primary" class="btn-block" (click)="Login()">
    Log in
  </button>


  <button mat-stroked-button color="primary" class="btn-block" (click)="GetCourses()">
    Get Courses
   </button>

  <div>{{courses | json}}</div>
  </div>
```

Frontend ● login.html

```
import { HttpClient } from '@angular/common/http';
import { Injectable } from '@angular/core';
import { map, Observable } from 'rxjs';

@Injectable({
  providedIn: 'root',
})
export class DataService {
  apiUrl = 'http://localhost:5240/api/';

  constructor(private httpClient: HttpClient) {}

  Login() {
    let user = new UserCredentials();
    return this.httpClient.post(`${this.apiUrl}Course/Login`, user);
  }

  Courses() {
    return this.httpClient.get<any>(`${this.apiUrl}Course/GetAllCourses`);
  }
}

/*********************************************
 * ******************
 * This is where you simulate the login by entering your own credentials created via the API Swagger UI
 * *******************
 **********************************************/
class UserCredentials {
  EmailAddress: string = 'test@gmail.com';
  Password: string = 'password123';
}
```

Frontend ● data.service

Application Security

Practical code

Frontend • auth-interceptor

```
import {
  HttpEvent,
  HttpHandler,
  HttpInterceptor,
  HttpRequest,
} from '@angular/common/http';
import { Injectable } from '@angular/core';
import { Observable } from 'rxjs';

@Injectable()
export class AuthInterceptor implements HttpInterceptor {
  intercept(
    req: HttpRequest<any>,
    next: HttpHandler,
  ): Observable<HttpEvent<any>> {
    if (localStorage.getItem('Token')) {
      const jwt = JSON.parse(localStorage.getItem('Token')!);
      const token = jwt.token;

      const cloned = req.clone({
        headers: req.headers.set('Authorization', 'Bearer ' + token),
      });

      return next.handle(cloned);
    } else {
      return next.handle(req);
    }
  }
}
```

Backend • Controller

```
using ApplicationSecurity_Backend.Models;
using ApplicationSecurity_Backend.ViewModels;
using Microsoft.AspNetCore.Authentication;
using Microsoft.AspNetCore.Authentication.JwtBearer;
using Microsoft.AspNetCore.Authorization;
using Microsoft.AspNetCore.Http;
using Microsoft.AspNetCore.Identity;
using Microsoft.AspNetCore.Mvc;
using Microsoft.IdentityModel.Tokens;
using System.IdentityModel.Tokens.Jwt;
using System.Security.Claims;
using System.Text;
```

Application Security

Practical code

```
namespace ApplicationSecurity_Backend.Controllers
{
   [Route("api/[controller]")]
   [ApiController]
   public class CourseController : ControllerBase
   {
      private readonly UserManager<AppUser> _userManager;
      private readonly RoleManager<IdentityRole> _roleManager;
      private readonly IRepository _repository;
      private readonly IUserClaimsPrincipalFactory<AppUser> _claimsPrincipalFactory;
      private readonly IConfiguration _configuration;

      public CourseController(UserManager<AppUser> userManager, RoleManager<IdentityRole>
roleManager, IUserClaimsPrincipalFactory<AppUser> claimsPrincipalFactory, IConfiguration
configuration, IRepository repository)
      {
         _userManager = userManager;
         _roleManager = roleManager;
         _claimsPrincipalFactory = claimsPrincipalFactory;
         _configuration = configuration;
         _repository = repository;
      }

      [HttpGet]
      [Route("GetAllCourses")]
      [Authorize(AuthenticationSchemes = JwtBearerDefaults.AuthenticationScheme)]
      public async Task<IActionResult> GetAllCoursesAsync()
      {
         try
         {
            var results = await _repository.GetAllCoursesAsync();
            return Ok(results);
         }
         catch (Exception)
         {
            return StatusCode(StatusCodes.Status500InternalServerError, "Internal Server Error,
please contact support");
         }
      }
```

Application Security

Practical code

```
[HttpPost]
    [Route("Register")]
    public async Task<IActionResult> Register(UserViewModel uvm)
    {
        var user = await _userManager.FindByIdAsync(uvm.emailaddress);

        if (user == null)
        {
            user = new AppUser
            {
                Id = Guid.NewGuid().ToString(),
                UserName = uvm.emailaddress,
                Email = uvm.emailaddress
            };

            var result = await _userManager.CreateAsync(user, uvm.password);

            if (result.Errors.Count() > 0) return
StatusCode(StatusCodes.Status500InternalServerError, "Internal Server Error. Please contact
support.");
        }
        else
        {
            return Forbid("Account already exists.");
        }

        return Ok();
    }

    [HttpPost]
    [Route("Login")]
    public async Task<ActionResult> Login(UserViewModel uvm)
    {
        var user = await _userManager.FindByNameAsync(uvm.emailaddress);

        if (user != null && await _userManager.CheckPasswordAsync(user, uvm.password))
        {
            try
            {
                return await GenerateJWTToken(user);
            }
            catch (Exception)
            {
                return StatusCode(StatusCodes.Status500InternalServerError, "Internal Server Error.
Please contact support.");
            }
        }
```

Application Security

Practical code

```
        else
        {
            return NotFound("Does not exist");
        }
    }

    [HttpGet]
    private async Task<ActionResult> GenerateJWTToken(AppUser user)
    {
        var role = await _userManager.GetRolesAsync(user);
        IdentityOptions _identityOptions = new IdentityOptions();
        // Create JWT Token
        var claims = new List<Claim>
        {
            new Claim(JwtRegisteredClaimNames.Sub, user.Email),
            new Claim(JwtRegisteredClaimNames.Jti, Guid.NewGuid().ToString()),
            new Claim(JwtRegisteredClaimNames.UniqueName, user.UserName),

        };

        if (role.Count() > 0)
        {
            claims.Add(new Claim(_identityOptions.ClaimsIdentity.RoleClaimType,
role.FirstOrDefault()));
        }

        var key = new
SymmetricSecurityKey(Encoding.UTF8.GetBytes(_configuration["Tokens:Key"]));
        var credentials = new SigningCredentials(key, SecurityAlgorithms.HmacSha256);

        var token = new JwtSecurityToken(
            _configuration["Tokens:Issuer"],
            _configuration["Tokens:Audience"],
            claims,
            signingCredentials: credentials,
            expires: DateTime.UtcNow.AddHours(3)
        );

        return Created("", new
        {
            token = new JwtSecurityTokenHandler().WriteToken(token),
            user = user.UserName
        });
    }
```

Application Security

Practical code

```
[HttpPost]
[Route("CreateRole")]
public async Task<IActionResult> CreateRole(string roleName)
{
    var role = await _roleManager.FindByNameAsync(roleName);
    if (role == null)
    {
        role = new IdentityRole
        {
            Id = Guid.NewGuid().ToString(),
            Name = roleName
        };

        var result = await _roleManager.CreateAsync(role);

        if (result.Errors.Count() > 0) return BadRequest(result.Errors);
    }
    else
    {
        return Forbid("Role already exists.");
    }

    return Ok();
}

[HttpPost]
[Route("AssignRole")]
public async Task<IActionResult> AssignRole(string emailAddress, string roleName)
{
    var user = await _userManager.FindByEmailAsync(emailAddress);
    if (user == null) return NotFound();

    var result = await _userManager.AddToRoleAsync(user, roleName);
    if (result.Succeeded) return Ok();

    return BadRequest(result.Errors);
}
```

Backend ● Controller

```
    [HttpGet]
    [Authorize(AuthenticationSchemes = "Bearer")]
    [Authorize(Roles = "Admin, Manager")]
    [Route("RoleTest")]
    public IActionResult RoleTest()
    {
        return Ok("You are an admin or manager!!!");
    }
  }
}
```

Backend ● Controller

Application Security

Practical code

```
using Microsoft.AspNetCore.Identity.EntityFrameworkCore;
using Microsoft.EntityFrameworkCore;

namespace ApplicationSecurity_Backend.Models
{
    public class AppDbContext:IdentityDbContext<AppUser>
    {

        public AppDbContext(DbContextOptions<AppDbContext> options) : base(options)
        {
        }

        public DbSet<Course> Courses { get; set; }

        protected override void OnModelCreating(ModelBuilder modelBuilder)
        {
            base.OnModelCreating(modelBuilder);

            modelBuilder.Entity<Course>()
                .HasData(
                new
                {
                    CourseId = 1,
                    Name = "AIM101",
                    Duration = "Semester",
                    Description = "Year 1, Semester 1. Academic Information Management"
                }
            );
            modelBuilder.Entity<Course>()
                .HasData(
                new
                {
                    CourseId = 2,
                    Name = "ALL121",
                    Duration = "Semester",
                    Description = "Year 1, Semester 2. Academic Literacy for IT"
                }
            );
            modelBuilder.Entity<Course>()
                .HasData(
                new
                {
                    CourseId = 3,
                    Name = "INF171",
                    Duration = "Year",
                    Description = "Year 1. Systems Analysis and Design"
                }
            );
            modelBuilder.Entity<Course>()
                .HasData(
                new
                {
                    CourseId = 4,
                    Name = "INF271",
                    Duration = "Year",
                    Description = "Year 2. Systems Analysis and Design"
                }
            );
```

Application Security

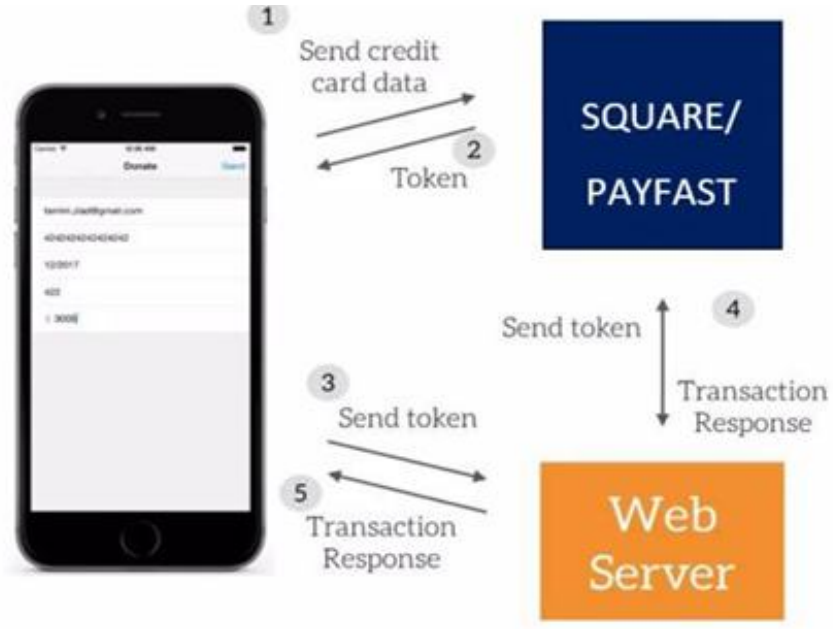
Practical code

```
        modelBuilder.Entity<Course>()
            .HasData(
            new
            {
                CourseId = 5,
                Name = "INF272",
                Duration = "Year",
                Description = "Year 2. Programming"
            }
        );
        modelBuilder.Entity<Course>()
            .HasData(
            new
            {
                CourseId = 6,
                Name = "INF214",
                Duration = "Semester",
                Description = "Year 2, Semester 1. Databases"
            }
        );
        modelBuilder.Entity<Course>()
            .HasData(
            new
            {
                CourseId = 7,
                Name = "INF315",
                Duration = "Semester",
                Description = "Year 3, Semester 1. Programming Management"
            }
        );
        modelBuilder.Entity<Course>()
            .HasData(
            new
            {
                CourseId = 8,
                Name = "INF324",
                Duration = "Semester",
                Description = "Year 3, Semester 2. IT Trends"
            }
        );
        modelBuilder.Entity<Course>()
            .HasData(
            new
            {
                CourseId = 9,
                Name = "INF354",
                Duration = "Semester",
                Description = "Year 3, Semester 1. Programming"
            }
        );
        modelBuilder.Entity<Course>()
            .HasData(
            new
            {
                CourseId = 10,
                Name = "INF370",
                Duration = "Year",
                Description = "Year 3. Project"
            }
        );
    }
  }
}
```

Advanced Concepts

- The technology merchants employ to receive customer debit or credit card payments is known as a payment gateway.
- The word covers both the online shopping cart payment processing portals and the actual card-reading hardware in real retail establishments.

How it works



CORS

CORS (Cross-Origin Resource Sharing) errors occur when a web page tries to make requests to a different domain (origin) than the one that served the web page. Browsers implement the same-origin policy to protect users, which restricts how web pages can make requests to different origins. CORS is a mechanism that allows servers to specify who can access their resources and which HTTP methods are allowed.

Understanding CORS and Same-Origin Policy

- Same-Origin Policy: This policy ensures that a web page can only request resources from the same origin it was served from. An origin is defined by the scheme (protocol), host (domain), and port. For example, https://example.com:8080 is different from http://example.com:8080 or https://example.com.
- CORS: CORS is a way to relax the same-origin policy. It allows servers to specify which origins can access their resources and which HTTP methods are permitted.

Advanced Concepts

CORS

CORS Headers

CORS involves the use of several HTTP headers that control access:

- Access-Control-Allow-Origin: Specifies which origins can access the resource. For example, Access-Control-Allow-Origin: https://example.com allows https://example.com to access the resource.
- Access-Control-Allow-Methods: Specifies which HTTP methods are allowed. For example, Access-Control-Allow-Methods: GET, POST.
- Access-Control-Allow-Headers: Specifies which headers can be used in the request.
- Access-Control-Allow-Credentials: Indicates whether the response to the request can be exposed when the credentials flag is true.

Example Scenario

Imagine you have a web application running on http://localhost:3000, and it tries to fetch data from an API at https://api.example.com. Without proper CORS headers set by the server at https://api.example.com, the browser will block the request and throw a CORS error.

How CORS Errors Happen

1. Preflight Requests: For certain types of requests (such as those using methods other than GET or POST, or custom headers), the browser first sends a preflight request using the OPTIONS method to check if the actual request is safe to send.
2. Missing or Incorrect Headers: If the server response does not include the necessary CORS headers (like Access-Control-Allow-Origin), or if the headers are incorrect, the browser will block the request and log a CORS error in the console.

ChatBots

- NLP: Natural Language Processing
- NLU: Natural Language Understanding
- NLG: Natural Language Generation

Key Files:

- NLU.yml– create intents
- Domain.yml– Register intents and Write response templates
- Stories.yml– Write stories using new intents and responses

Terminology

- Narrow assistant – defined skill-set
- Intents - are the goals or meaning the user is trying to convey
- Entities – are important keywords, that you want to capture, that the models extract from users message
- Custom Action - getting the assistant to do something for you. E.g., connect to a database, posting data to an API, send data to a spreadsheet
- Forms –uses slots to store values

Assignment 2

Frontend

```
import { ComponentFixture, TestBed, waitForAsync } from '@angular/core/testing';
import { IonicModule } from '@ionic/angular';

import { BannerComponent } from './banner.component';

describe('BannerComponent', () => {
  let component: BannerComponent;
  let fixture: ComponentFixture<BannerComponent>;

  beforeEach(waitForAsync(() => {
    TestBed.configureTestingModule({
      declarations: [BannerComponent],
      imports: [IonicModule.forRoot()],
    }).compileComponents();

    fixture = TestBed.createComponent(BannerComponent);
    component = fixture.componentInstance;
    fixture.detectChanges();
  }));

  it('should create', () => {
    expect(component).toBeTruthy();
  });
});
```

Components ● banner.ts

```
<ion-row class="borderBottom" *ngIf="bannerImages?.length > 0">
  <div>
    <swiper-container
    [modules]="swiperModules"
    [slidesPerView]="1.2"
    [spaceBetween]="20"
    [pagination]="{clickable: true, dynamicBullets: true}">
    <swiper-slide *ngFor="let image of bannerImages">
      <img [src]="image?.banner" />
    </swiper-slide>
  </swiper-container>
  </div>
</ion-row>
```

Components ● banner.html

```angular2html
import { Component, OnInit, Input } from '@angular/core';
import { Restaurant } from 'src/app/models/restaurant.model';

@Component({
  selector: 'app-restaurant',
  templateUrl: './restaurant.component.html',
  styleUrls: ['./restaurant.component.scss'],
})
export class RestaurantComponent implements OnInit {
  @Input() restaurant: Restaurant;

  constructor() {}

  ngOnInit() {}

  getCuisine(cuisine) {
    return cuisine.join(', ');
  }
}
```

Components ● restaurant.ts

```angular2html
<ion-item lines="none">
  <ion-thumbnail slot="start">
    <img [src]="restaurant?.cover ? restaurant.cover : 'assets/imgs/1.jpg'" />
  </ion-thumbnail>
  <ion-label>
    <h4>{{restaurant?.name}}</h4>
    <ion-text color="medium">
      <p class="pStyle">
        {{getCuisine(restaurant?.cuisines)}}
      </p>
    </ion-text>
    <span>
      <ion-icon name="star"></ion-icon>
      {{restaurant?.rating}} .
    </span>
    {{restaurant?.delivery_time}} mins . R{{restaurant?.price}} for two
    <ion-text color="tertiary" *ngIf="restaurant?.distance && restaurant?.distance != 0">
      <p class="distance">
        {{restaurant?.distance | number: '0.0-2'}} kms away
      </p>
    </ion-text>
  </ion-label>
</ion-item>
```

Components ● restaurant.html

```angular2html
import { Component, OnInit, Input } from '@angular/core';
import { Restaurant } from 'src/app/models/restaurant.model';

@Component({
  selector: 'app-restaurant-detail',
  templateUrl: './restaurant-detail.component.html',
  styleUrls: ['./restaurant-detail.component.scss'],
})
export class RestaurantDetailComponent implements OnInit {
  @Input() data: Restaurant;
  @Input() isLoading;

  constructor() {}

  ngOnInit() {}

  getCuisine(cuisine) {
    return cuisine.join(', ');
  }
}
```

Components ● restaurant-detail.ts

```angular2html
<ion-grid class="borderBottom" *ngIf="isLoading">
  <ion-item lines="none" class="ion-padding-horizontal dashedBorderBottom">
    <ion-label>
      <h1>
        <ion-skeleton-text style="width: 50%" animated></ion-skeleton-text>
      </h1>
      <p>
        <ion-skeleton-text style="width: 60%" animated></ion-skeleton-text>
      </p>
      <p>
        <ion-skeleton-text style="width: 80%" animated></ion-skeleton-text>
      </p>
    </ion-label>
  </ion-item>
  <ion-row>
    <ion-col size="4">
      <ion-label class="ion-text-center">
        <p><ion-skeleton-text animated style="width: 30%"></ion-skeleton-text></p>
        <p><ion-skeleton-text animated style="width: 35%"></ion-skeleton-text></p>
      </ion-label>
    </ion-col>
    <ion-col size="4">
      <ion-label class="ion-text-center">
        <p><ion-skeleton-text animated style="width: 30%"></ion-skeleton-text></p>
        <p><ion-skeleton-text animated style="width: 35%"></ion-skeleton-text></p>
      </ion-label>
    </ion-col>
    <ion-col size="4">
      <ion-label class="ion-text-center">
        <p><ion-skeleton-text animated style="width: 30%"></ion-skeleton-text></p>
        <p><ion-skeleton-text animated style="width: 35%"></ion-skeleton-text></p>
      </ion-label>
    </ion-col>
```

Components ● restaurant-detail.html

```
<ion-grid *ngIf="data?.name && !isLoading" class="borderBottom">
  <ion-row class="ion-padding-horizontal dashedBorderBottom">
    <ion-label>
      <h1>{{data?.name}}</h1>
      <ion-text *ngIf="data?.cuisines"><p>{{getCuisine(data.cuisines)}}</p></ion-text>
      <p>{{data?.address}}</p>
    </ion-label>
  </ion-row>
  <ion-row>
    <ion-col size="4">
      <ion-label class="ion-text-center">
        <ion-text>
          <p>
            <ion-icon name="star"></ion-icon>{{data?.rating}}
          </p>
        </ion-text>
        <p>Ratings</p>
      </ion-label>
    </ion-col>
    <ion-col size="4">
      <ion-label class="ion-text-center">
        <ion-text>
          <p>
            {{data?.delivery_time}} MINS
          </p>
        </ion-text>
        <p>Delivery Time</p>
      </ion-label>
    </ion-col>
    <ion-col size="4">
      <ion-label class="ion-text-center">
        <ion-text>
          <p>
            R{{data?.price}}
          </p>
        </ion-text>
        <p>For Two</p>
      </ion-label>
    </ion-col>
  </ion-row>
</ion-grid>
```

```angular2html
import { Component, OnInit, Input, Output, EventEmitter } from '@angular/core';
import { Item } from 'src/app/models/item.model';

@Component({
  selector: 'app-cart-item',
  templateUrl: './cart-item.component.html',
  styleUrls: ['./cart-item.component.scss'],
})
export class CartItemComponent implements OnInit {
  @Input() item: Item;
  @Input() index: any;
  @Output() add: EventEmitter<any> = new EventEmitter();
  @Output() minus: EventEmitter<any> = new EventEmitter();

  constructor() {}

  ngOnInit() {}

  quantityPlus() {
    this.add.emit(this.index);
  }

  quantityMinus() {
    this.minus.emit(this.index);
  }
}
```

Components ● cart-item.ts

```angular2html
<ion-item>
  <ion-text color="success">
    <span class="dotStyle">&#9679;</span>
  </ion-text>
  <ion-label class="ion-text-wrap">
    <h5>{{item?.name}}</h5>
  </ion-label>
  <ion-row>
    <ion-icon color="success" name="remove-outline" (click)="quantityMinus()"></ion-icon>
    <ion-text class="ion-text-center quantity" color="success">{{item?.quantity}}</ion-text>
    <ion-icon color="success" name="add-outline" (click)="quantityPlus()"></ion-icon>
  </ion-row>
  <ion-text slot="end" color="dark" class="price ion-text-right">
    R{{(item?.price * item.quantity) | number: '0.2-2'}}
  </ion-text>
</ion-item>
```

Components ● cart-item.html

```angular2html
import { Component, OnInit, Input, Output, EventEmitter } from '@angular/core';
import { Order } from 'src/app/models/order.model';

@Component({
  selector: 'app-orders',
  templateUrl: './orders.component.html',
  styleUrls: ['./orders.component.scss'],
})
export class OrdersComponent implements OnInit {
  @Input() order: Order;
  @Output() reorder: EventEmitter<any> = new EventEmitter();
  @Output() help: EventEmitter<any> = new EventEmitter();

  constructor() {}

  ngOnInit() {}

  reorderItem() {
    this.reorder.emit(this.order);
  }

  getHelp() {
    this.help.emit(this.order);
  }
}
```

Components ● orders.ts

```angular2html
<ion-item-group class="borderBottom">
  <ion-item lines="none" class="dashedBorderBottom">
    <ion-label>
      {{order?.restaurant?.name}}
      <p>{{order?.restaurant?.address}}</p>
      <ion-text>
        <p>
          R{{order?.grandTotal}}
          <ion-icon name="chevron-forward-outline"></ion-icon>
        </p>
      </ion-text>
    </ion-label>
    <ion-note slot="end" color="dark">
      {{order?.status}}
      <ion-icon name="checkmark-circle-outline" color="success"></ion-icon>
    </ion-note>
  </ion-item>
  <ion-item lines="none">
    <ion-label>
      <ion-text color="dark" *ngFor="let item of order?.order">
        <p>{{item?.name}} x {{item?.quantity}}</p>
      </ion-text>
      <p class="time">{{order?.time}}</p>
      <ion-row>
        <ion-col size="6">
          <ion-button fill="outline" expand="block" color="primary"
(click)="reorderItem()">REORDER</ion-button>
        </ion-col>
        <ion-col size="6">
          <ion-button fill="outline" expand="block" color="dark"
(click)="getHelp()">GET HELP</ion-button>
        </ion-col>
      </ion-row>
    </ion-label>
  </ion-item>
</ion-item-group>
```

Components ● orders.html

```
import { Component, OnInit, Input, Output, EventEmitter } from '@angular/core';
import { Item } from 'src/app/models/item.model';

@Component({
  selector: 'app-item',
  templateUrl: './item.component.html',
  styleUrls: ['./item.component.scss'],
})
export class ItemComponent implements OnInit {
  @Input() item: Item;
  @Input() index;
  @Output() add: EventEmitter<Item> = new EventEmitter();
  @Output() minus: EventEmitter<Item> = new EventEmitter();

  constructor() {}

  ngOnInit() {}

  quantityPlus() {
    this.add.emit(this.item);
  }

  quantityMinus() {
    this.minus.emit(this.item);
  }
}
```

Components ● item.ts

```
<ion-item>
  <ion-thumbnail slot="start">
    <img [src]="item?.cover ? item.cover : 'assets/imgs/1.jpg'" />
  </ion-thumbnail>
  <ion-label>
    <!-- <h4>{{item?.name}}</h4> -->
    {{item?.name}}
    <p class="ion-text-wrap">{{item?.desc}}</p>
    <ion-text><p>R{{item?.price | number: '0.2-2'}}</p></ion-text>
  </ion-label>
  <ion-row *ngIf="!item?.quantity || item?.quantity <= 0"
(click)="quantityPlus()">
    <ion-text class="add" color="success">Add</ion-text>
  </ion-row>
  <ion-row *ngIf="item?.quantity > 0">
    <ion-icon color="success" name="remove-outline" (click)="quantityMinus()"></
ion-icon>
    <ion-text class="ion-text-center quantity" color="success">{{item?.quantity}}
</ion-text>
    <ion-icon color="success" name="add-outline" (click)="quantityPlus()"></ion-
icon>
  </ion-row>
</ion-item>
```

Components ● item.html

```
import { Injectable } from '@angular/core';
import { BehaviorSubject } from 'rxjs';
import { Address } from 'src/app/models/address.model';
import { ApiService } from '../api/api.service';

@Injectable({
  providedIn: 'root',
})
export class AddressService {
  private _addresses = new BehaviorSubject<Address[]>([]);

  get addresses() {
    return this._addresses.asObservable();
  }

  constructor(private api: ApiService) {}

  getAddresses() {
    try {
      //user id
      let allAddress: Address[] = this.api.addresses;
      console.log(allAddress);
      this._addresses.next(allAddress);
    } catch (e) {
      console.log(e);
      throw e;
    }
  }

  addAddress(param) {
    param.id = 'address1';
    param.user_id = 'user1';
    const currentAddresses = this._addresses.value;
    currentAddresses.push(
      new Address(
        param.id,
        param.user_id,
        param.title,
        param.address,
        param.landmark,
        param.house,
        param.lat,
        param.lng,
      ),
    );
    this._addresses.next(currentAddresses);
  }

  updateAddress(id, param) {
    param.id = id;
    let currentAddresses = this._addresses.value;
    const index = currentAddresses.findIndex((x) => x.id == id);
    currentAddresses[index] = new Address(
      id,
      param.user_id,
      param.title,
      param.address,
      param.landmark,
      param.house,
      param.lat,
      param.lng,
    );
    this._addresses.next(currentAddresses);
  }

  deleteAddress(param) {
    let currentAddresses = this._addresses.value;
    currentAddresses = currentAddresses.filter((x) => x.id != param.id);
    this._addresses.next(currentAddresses);
  }
}
```

```
import { Injectable } from '@angular/core';
import { Address } from 'src/app/models/address.model';
import { Category } from 'src/app/models/category.model';
import { Item } from 'src/app/models/item.model';
import { Order } from 'src/app/models/order.model';
import { Restaurant } from 'src/app/models/restaurant.model';

@Injectable({
  providedIn: 'root',
})
export class ApiService {
  banners = [
    { banner: 'assets/imgs/3.jpg' },
    { banner: 'assets/imgs/2.jpg' },
    { banner: 'assets/imgs/1.jpg' },
  ];

  restaurants: Restaurant[] = [
    {
      uid: '12wefdss',
      cover: 'assets/imgs/1.jpg',
      name: 'Jollof of Africa',
      short_name: 'Jollof of Africa',
      cuisines: ['African'],
      rating: 5,
      delivery_time: 25,
      distance: 2.38,
      price: 150,
    },
    {
      uid: '12wefdefsdss',
      cover: 'assets/imgs/2.jpg',
      name: 'Spice - The Indian Kitchen',
      short_name: 'Spice - The Indian Kitchen',
      cuisines: ['Indian'],
      rating: 4.0,
      delivery_time: 5,
      distance: 1.5,
      price: 100,
    },
    {
      uid: '12wefdssrete',
      cover: 'assets/imgs/3.jpg',
      name: 'Boujee Shisanyama',
      short_name: 'Boujee Shisanyama',
      cuisines: ['African'],
      rating: 4.7,
      delivery_time: 25,
      distance: 2.5,
      price: 110,
    },
  ];

  allRestaurants: Restaurant[] = [
    {
      uid: '12wefdss',
      cover: 'assets/imgs/1.jpg',
      name: 'Jollof of Africa',
      short_name: 'Jollof of Africa',
      cuisines: ['African'],
      rating: 5,
      delivery_time: 25,
      distance: 2.38,
      price: 150,
    },
```

```
  {
    uid: '12wefdefsdss',
    cover: 'assets/imgs/2.jpg',
    name: 'Spice - The Indian Kitchen',
    short_name: 'Spice - The Indian Kitchen',
    cuisines: ['Indian'],
    rating: 4.0,
    delivery_time: 5,
    distance: 1.5,
    price: 100,
  },
  {
    uid: '12wefdssrete',
    cover: 'assets/imgs/3.jpg',
    name: 'Boujee Shisanyama',
    short_name: 'Boujee Shisanyama',
    cuisines: ['African'],
    rating: 4.7,
    delivery_time: 25,
    distance: 2.5,
    price: 110,
  },
];

restaurants1: Restaurant[] = [
  {
    uid: '12wefdss',
    cover: 'assets/imgs/1.jpg',
    name: 'Jollof of Africa',
    short_name: 'Jollof of Africa',
    address: 'Francis Baard Street',
    cuisines: ['African'],
    rating: 5,
    delivery_time: 25,
    distance: 2.38,
    price: 150,
  },
  {
    uid: '12wefdefsdss',
    cover: 'assets/imgs/2.jpg',
    name: 'Spice - The Indian Kitchen',
    short_name: 'Spice - The Indian Kitchen',
    cuisines: ['Indian'],
    rating: 4.0,
    delivery_time: 5,
    address: 'Lynnwood Ridge',
    distance: 1.5,
    price: 100,
  },
  {
    uid: '12wefdssrete',
    cover: 'assets/imgs/3.jpg',
    name: 'Boujee Shisanyama',
    short_name: 'Boujee Shisanyama',
    cuisines: ['African'],
    rating: 4.7,
    delivery_time: 25,
    address: 'Arcadia Street, Hatfield',
    distance: 2.5,
    price: 110,
  },
];
```

```
categories: Category[] = [
    {
      id: 'e0',
      name: 'Indian',
      uid: '12wefdefsdss',
    },
    {
      id: 'e00',
      name: 'African',
      uid: '12wefdss',
    },
    {
      id: 'e01',
      name: 'African',
      uid: '12wefdssrete',
    },
  ];

  allItems: Item[] = [
    {
      category_id: 'e0',
      cover: 'assets/imgs/2.jpg',
      desc: 'Great in taste',
      id: 'i1',
      name: 'Bunny Chow',
      price: 50,
      rating: 0,
      status: true,
      uid: '12wefdefsdss',
      variation: false,
      veg: false,
    },
    {
      category_id: 'e00',
      cover: 'assets/imgs/1.jpg',
      desc: 'Great in taste',
      id: 'i1',
      name: 'Jollof Rice',
      price: 75,
      rating: 0,
      status: true,
      uid: '12wefdss',
      variation: false,
      veg: false,
    },
    {
      category_id: 'e01',
      cover: 'assets/imgs/3.jpg',
      desc: 'Great in taste',
      id: 'i1',
      name: 'Shisanyama and Pap',
      price: 55,
      rating: 0,
      status: true,
      uid: '12wefdssrete',
      variation: false,
      veg: false,
    },
  ];
```

```
  addresses: Address[] = [
    {
      address: 'University of Pretoria',
      house: '4th Floor',
      id: '7Kox63KlggTvV7ebRKar',
      landmark: 'Informatorium',
      lat: 26.1830738,
      lng: 91.74049769999999,
      title: 'Work',
      user_id: '1',
    },
    {
      address: 'Department of Informatics',
      house: '5th Floor',
      id: '8Kox63KlggTvV7ebRKar',
      landmark: 'University of Pretoria',
      lat: 26.1830738,
      lng: 91.74049769999999,
      title: 'Home',
      user_id: '1',
    },
  ];

  orders: Order[] = [
    {
      address: {
        address: 'University of Pretoria, Infomatorium',
        house: 'dsgd',
        id: 'cLQdnS8YXk5HTDfM3UQC',
        landmark: 'fdgs',
        lat: 26.108991978867923,
        lng: 91.79069981213378,
        title: 'yui',
        user_id: '1',
      },
      deliveryCharge: 20,
      grandTotal: 200.0,
      id: '5aG0RsPuze8NX00B7uRP',
      order: [
        {
          category_id: 'e0',
          cover: 'assets/imgs/2.jpg',
          desc: 'Great in taste',
          id: 'i2',
          name: 'Bunny Chow',
          price: 200,
          rating: 0,
          status: true,
          uid: '12wefdefsdss',
          variation: false,
          veg: true,
          quantity: 4,
        },
      ],
      paid: 'COD',
      restaurant: {
        uid: '12wefdefsdss',
        cover: 'assets/imgs/2.jpg',
        name: 'Spice - The Indian Kitchen',
        short_name: 'Spice - The Indian Kitchen',
        cuisines: ['Indian'],
        rating: 4.0,
        delivery_time: 5,
        distance: 1.5,
        price: 100,
      },
      restaurant_id: '12wefdefsdss',
      status: 'Delivered',
      time: 'May 6, 2024 11:44 AM',
      total: 200.0,
      user_id: '1',
    },
  ];

  constructor() {}
}
```

```
import { Injectable } from '@angular/core';
import { Router } from '@angular/router';
import { BehaviorSubject } from 'rxjs';
import { Cart } from 'src/app/models/cart.model';
import { Item } from 'src/app/models/item.model';
import { Order } from 'src/app/models/order.model';
import { Restaurant } from 'src/app/models/restaurant.model';
import { GlobalService } from '../global/global.service';
import { StorageService } from '../storage/storage.service';

@Injectable({
  providedIn: 'root',
})
export class CartService {
  model = {} as Cart;
  deliveryCharge = 20;
  priv

  async orderToCart(order: Order) {
    console.log('order: ', order);
    const data = {
      restaurant: order.restaurant,
      items: order.order,
    };
    this.model = data;
    await this.calculate();
    this.saveCart();
    console.log('model: ', this.model);
    this._cart.next(this.model);
    this.router.navigate(['/', 'tabs', 'restaurants', order.restaurant_id]);
  }

  async quantityPlus(index, items?: Item[], restaurant?: Restaurant) {
    try {
      if (items) {
        console.log('model: ', this.model);
        this.model.items = [...items];
      }
      if (restaurant) {
        // this.model.restaurant = {};
        this.model.restaurant = restaurant;
      }
      console.log('q plus: ', this.model.items[index]);
      // this.model.items[index].quantity += 1;
      if (
        !this.model.items[index].quantity ||
        this.model.items[index].quantity == 0
      ) {
        this.model.items[index].quantity = 1;
      } else {
        this.model.items[index].quantity += 1; // this.model.items[index].quantity
= this.model.items[index].quantity + 1
      }
      await this.calculate();
      this._cart.next(this.model);
    } catch (e) {
      console.log(e);
      throw e;
    }
  }
```

```
 async quantityMinus(index, items?: Item[]) {
    try {
      if (items) {
        console.log('model: ', this.model);
        this.model.items = [...items];
      }
      console.log('item: ', this.model.items[index]);
      if (
        this.model.items[index].quantity &&
        this.model.items[index].quantity !== 0
      ) {
        this.model.items[index].quantity -= 1; // this.model.items[index].quantity =
this.model.items[index].quantity - 1
      } else {
        this.model.items[index].quantity = 0;
      }
      await this.calculate();
      this._cart.next(this.model);
    } catch (e) {
      console.log(e);
      throw e;
    }
  }

  async calculate() {
    let item = this.model.items.filter((x) => x.quantity > 0);
    this.model.items = item;
    this.model.totalPrice = 0;
    this.model.totalItem = 0;
    this.model.deliveryCharge = 0;
    this.model.grandTotal = 0;
    item.forEach((element) => {
      this.model.totalItem += element.quantity;
      // this.model.totalPrice += (parseFloat(element.price) *
parseFloat(element.quantity));
      this.model.totalPrice += element.price * element.quantity;
    });
    this.model.deliveryCharge = this.deliveryCharge;
    // this.model.totalPrice = parseFloat(this.model.totalPrice).toFixed(2);
    // this.model.grandTotal = (parseFloat(this.model.totalPrice) +
parseFloat(this.model.deliveryCharge)).toFixed(2);
    this.model.grandTotal = this.model.totalPrice + this.model.deliveryCharge;
    if (this.model.totalItem == 0) {
      this.model.totalItem = 0;
      this.model.totalPrice = 0;
      this.model.grandTotal = 0;
      await this.clearCart();
      this.model = {} as Cart;
    }
    console.log('cart: ', this.model);
  }

  async clearCart() {
    this.global.showLoader();
    await this.storage.removeStorage('cart');
    this._cart.next(null);
    this.global.hideLoader();
  }

  saveCart(model?) {
    if (model) this.model = model;
    this.storage.setStorage('cart', JSON.stringify(this.model));
    // this._cart.next(this.model);
  }
}
```

```angular2html
import { Injectable } from '@angular/core';
import {
  AlertController,
  LoadingController,
  ModalController,
  ToastController,
} from '@ionic/angular';

@Injectable({
  providedIn: 'root',
})
export class GlobalService {
  isLoading: boolean = false;

  constructor(
    private alertCtrl: AlertController,
    private toastCtrl: ToastController,
    private loadingCtrl: LoadingController,
    private modalCtrl: ModalController,
  ) {}

  setLoader() {
    this.isLoading = !this.isLoading;
  }

  showAlert(message: string, header?, buttonArray?) {
    this.alertCtrl
      .create({
        header: header ? header : 'Authentication failed',
        message: message,
        buttons: buttonArray ? buttonArray : ['Okay'],
      })
      .then((alertEl) => alertEl.present());
  }

  async showToast(msg, color, position, duration = 3000) {
    const toast = await this.toastCtrl.create({
      message: msg,
      duration: duration,
      color: color,
      position: position,
    });
    toast.present();
  }

  errorToast(msg?, duration = 4000) {
    this.showToast(
      msg ? msg : 'No Internet Connection',
      'danger',
      'bottom',
      duration,
    );
  }
```

```
  successToast(msg) {
    this.showToast(msg, 'success', 'bottom');
  }

  showLoader(msg?, spinner?) {
    // this.isLoading = true;
    if (!this.isLoading) this.setLoader();
    return this.loadingCtrl
      .create({
        message: msg,
        spinner: spinner ? spinner : 'bubbles',
      })
      .then((res) => {
        res.present().then(() => {
          if (!this.isLoading) {
            res.dismiss().then(() => {
              console.log('abort presenting');
            });
          }
        });
      })
      .catch((e) => {
        console.log('show loading error: ', e);
      });
  }

  hideLoader() {
    // this.isLoading = false;
    if (this.isLoading) this.setLoader();
    return this.loadingCtrl
      .dismiss()
      .then(() => console.log('dismissed'))
      .catch((e) => console.log('error hide loader: ', e));
  }

  async createModal(options) {
    const modal = await this.modalCtrl.create(options);
    await modal.present();
    const { data } = await modal.onWillDismiss();
    console.log(data);
    if (data) return data;
  }

  modalDismiss(val?) {
    let data: any = val ? val : null;
    console.log('data', data);
    this.modalCtrl.dismiss(data);
  }

  getIcon(title) {
    const name = title.toLowerCase();
    switch (name) {
      case 'home':
        return 'home-outline';
      case 'work':
        return 'briefcase-outline';
      default:
        return 'location-outline';
    }
  }
}
```

```
import { Injectable } from '@angular/core';
import { BehaviorSubject } from 'rxjs';
import { Order } from 'src/app/models/order.model';
import { ApiService } from '../api/api.service';

@Injectable({
  providedIn: 'root',
})
export class OrderService {
  private _orders = new BehaviorSubject<Order[]>([]);

  get orders() {
    return this._orders.asObservable();
  }

  constructor(private api: ApiService) {}

  getOrders() {
    try {
      const orders = this.api.orders;
      console.log('orders', orders);
      this._orders.next(orders);
    } catch (e) {
      throw e;
    }
  }

  placeOrder(param) {
    try {
      param.user_id = '1';
      // param.order = JSON.stringify(param.order);
      // param.order = JSON.parse(param.order);
      param.id = '5aG0RsPuze8NX00B7uE2';
      // let currentOrders = this._orders.value;
      console.log('latest order: ', param);
      let currentOrders: Order[] = [];
      currentOrders.push(
        new Order(
          param.address,
          param.restaurant,
          param.restaurant_id,
          param.order,
          param.total,
          param.grandTotal,
          param.deliveryCharge,
          param.status,
          param.time,
          param.paid,
          param.id,
          param.user_id,
          param.instruction,
        ),
      );
      console.log('latest order: ', currentOrders);
      currentOrders = currentOrders.concat(this._orders.value);
      console.log('orders: ', currentOrders);
      this._orders.next(currentOrders);
    } catch (e) {
      throw e;
    }
  }
}
```

```
import { Injectable } from '@angular/core';
import { BehaviorSubject } from 'rxjs';
import { Order } from 'src/app/models/order.model';
import { ApiService } from '../api/api.service';

@Injectable({
  providedIn: 'root',
})
export class OrderService {
  private _orders = new BehaviorSubject<Order[]>([]);

  get orders() {
    return this._orders.asObservable();
  }

  constructor(private api: ApiService) {}

  getOrders() {
    try {
      const orders = this.api.orders;
      console.log('orders', orders);
      this._orders.next(orders);
    } catch (e) {
      throw e;
    }
  }

  placeOrder(param) {
    try {
      param.user_id = '1';
      // param.order = JSON.stringify(param.order);
      // param.order = JSON.parse(param.order);
      param.id = '5aG0RsPuze8NX00B7uE2';
      // let currentOrders = this._orders.value;
      console.log('latest order: ', param);
      let currentOrders: Order[] = [];
      currentOrders.push(
        new Order(
          param.address,
          param.restaurant,
          param.restaurant_id,
          param.order,
          param.total,
          param.grandTotal,
          param.deliveryCharge,
          param.status,
          param.time,
          param.paid,
          param.id,
          param.user_id,
          param.instruction,
        ),
      );
      console.log('latest order: ', currentOrders);
      currentOrders = currentOrders.concat(this._orders.value);
      console.log('orders: ', currentOrders);
      this._orders.next(currentOrders);
    } catch (e) {
      throw e;
    }
  }
}
```

```
import { Injectable } from '@angular/core';
import { Preferences } from '@capacitor/preferences';

@Injectable({
  providedIn: 'root',
})
export class StorageService {
  constructor() {}

  setStorage(key, value) {
    Preferences.set({ key: key, value: value });
  }

  getStorage(key) {
    return Preferences.get({ key: key });
  }

  removeStorage(key) {
    Preferences.remove({ key: key });
  }

  clearStorage() {
    Preferences.clear();
  }
}
```

//app/

```
import { Component, OnInit, OnDestroy } from '@angular/core';
import { Subscription } from 'rxjs';
import { ActivatedRoute, Router } from '@angular/router';
import { CartService } from 'src/app/services/cart/cart.service';
import { CoreService } from 'src/app/services/core/core.service';

@Component({
  selector: 'app-account',
  templateUrl: './account.page.html',
  styleUrls: ['./account.page.scss'],
})
export class AccountPage implements OnInit, OnDestroy {
  profile: any = {};
  isLoaded: boolean = false;
  orders: Order[] = [];
  routeSub: Subscription;

  constructor(
    private coreService: CoreService,
    private cartService: CartService,
  ) {}

  ngOnInit() {
    this.coreService.crUserOrders.subscribe(
      (orders) => {
        console.log('orders data: ', orders);
        this.orders = orders;
        // this.isLoaded = true;
        if (!this.orders.length) {
          // else {
            // this.isLoaded = false;
            // this.orders = this.orders.filter(o => o.id != order.id);
            // } else if(order.updated) {
            // this.orders = this.orders.map(o => o.id == order.id);
            // this.orders[index] = order;
            // } else {
            // this.orders = this.orders.concat(order);
            // }
        }
      },
      (e) => {
        console.log(e);
      }
    );
    this.getOrdersFn();
  }

  async getOrdersFn() {
    this.isLoaded = false;
    let[ordersCollection] = [];
    this.profile = {
      name: 'Test User',
      phone: '9876 543 210',
      email: 'timothy.adm1@yahoo.co.in',
    };
    ordersCollection = this.coreService.getOrdersFn();
    this.isLoaded = false;
    } catch(e) {
    } finally {
    }
    this.orders = ordersCollection;
    console.log(this.orders);
    this.isLoaded = true;
  }

  logout() {}

  async reorder(order: Order) {
    console.log('reorder');
    let data = await this.cartService.getCart();
    console.log('data: ', data);
    if (!data.length) {
      this.cartService.clearCleanCart(null, null, null, order);
    } else {
      this.cartService.reorderCart(order);
    }
  }

  getHelp(order) {
    console.log(order);
  }

  ngOnDestroy() {
    if (this.routeSub) this.routeSub.unsubscribe();
  }
}

import { Component, OnInit, OnDestroy } from '@angular/core';
import { Subscription } from 'rxjs';
import { ActivatedRoute, Router } from '@angular/router';
import { AddressesPage } from 'src/app/modals/addresses.modal/';
import { AddressEdit } from 'src/app/modals/addresses.modal/';
import { GlobalSpinnerSvc } from 'src/app/services/global/global.service';

@Component({
  selector: 'app-address',
  templateUrl: './address.page.html',
  styleUrls: ['./address.page.scss'],
}
export class AddressPage implements OnInit, OnDestroy {
  isLoading: boolean;
  addresses: Address[] = [];
  routeSub: Subscription;
  model = {
    title: 'No Addresses added yet',
    icon: 'location-outline',
  };

  constructor(
    private global: GlobalService,
    private addressSvc: AddressSvc,
  )

  ngOnInit() {
    this.addressSvc.crAddresses.subscribe(addresses => {
      console.log('addresses: ', address);
      this.addresses = address;
    });
    this.getAddressesFn();
  }

  async getAddressesFn() {
    this.isLoading = true;
    this.global.showLoader();
    setTimeout(async () => {
      await this.addressSvc.getAddresses();
      this.global.hideLoader();
      this.isLoading = false;
      this.global.hideLoader();
    }, 500);
  }

  getIcon(title) {
    return this.global.getIconTitle(title);
  }

  editAddress(address) {}

  deleteAddress(address) {
    console.log('address: ', address);
    this.global.showAlert({
      msg: 'Are you sure want to delete this address?',
      confirm: [
        {
          text: 'No',
          role: 'cancel',
          handler: () => {
            console.log('cancel');
            return;
          },
        },
        {
          text: 'Yes',
          handler: async () => {
            this.global.showLoader();
            this.addressSvc.deleteAddress(address);
            this.global.hideLoader();
          },
        }
      ],
    });
  }

  ngOnDestroy() {
    if (this.addressRouteSub) this.addressRouteSub.unsubscribe();
  }
}

import { Component, OnInit, ViewChild } from '@angular/core';
import { Router } from '@angular/router';
import { IonContent, NavController } from '@ionic/angular';
import { Subscription } from 'rxjs';
import { Address } from 'src/app/models/address.model';
import { AddressSvc } from 'src/app/services/address/address.service';
import { CartService } from 'src/app/services/cart/cart.service';
import { CartSvcSvc } from 'src/app/services/cart/cart.service';
import { GlobalSpinnerSvc } from 'src/app/services/global/global.service';
import { CoreSvc } from 'src/app/services/core/core.service';

@Component({
  selector: 'app-cart',
  templateUrl: './cart.page.html',
  styleUrls: ['./cart.page.scss'],
}
export class CartPage implements OnInit {
  @ViewChild('ionContent', { static: false }) content: IonContent;
  addresses: Array;
  will any;
  isLoaded = false;
  deliveryCharge = 0;
  instructions: any;
  cartSub: Subscription;
  cartSub: Subscription;

  constructor(
    private navCtrl: NavController,
    private router: Router,
    private addressSvc: AddressSvc,
    private global: GlobalService,
    private cartService: CartService,
  )

  ngOnInit() {
    this.cartSub = this.cartService.cart.subscribe(cart => {
      console.log('cart page: ', cart);
      this.model = cart;
      this.addressSvc.this.location = {} as Address;
      console.log('cart page model: ', this.model);
    });
    this.getDefaultFn();
  }

  async getDefaultFn() {
    await this.checkIfEl();
    // this.location = {
    // name: 'Test User',
    // id: 77.180272,
    // address: 'Karol Bagh, New Delhi',
    // user_id: 1ooo1
    ...
    this.location = new Address(
      {name: 'Test User',
      address: {
        'address 1': 'Karol Bagh, New Delhi',
        ...
        ...
        ID:180272,
        77.180272,
        ...
      }
    );
    await this.cartSvcSvc.getCartSafe();
    ...
    checkIfEl() {
      let elt: any = this.content.el.nativeElement;
      console.log('elt: ', elt);
      console.log('this.content.el.length - 2, 2); // /tabs/cart/el.length - 1: 1
      this.elEnd = elt[elt[elt.length - 1].shadowRoot];
      console.log('elEnd: ', this.elEnd);
      console.log('this.elEnd: ', this.elEnd);
      this.will = elEnd;
      console.log(this.will);
    }

    getPreviousEl() {
      return this.will.prev();
    }

    quantityPlus(index) {
      this.cartSvcSvc.quantityPlus(index);
    }

    quantityMinus(index) {
      this.cartSvcSvc.quantityMinus(index);
    }

    addNewAddress() {}

    changeAddressFn() {}

    async makePayment() {
      try {
        console.log('model: ', this.model);
        let order = {
          restaurant_id: this.model.restaurant.id,
          instruction: this.instruction ? this.instruction : '',
          delivery_charge: this.deliveryCharge,
          order: this.model.items, //JSON.stringify(this.model.items),
          time: moment().format('I:I:I'),
          date: moment().format('L'),
          total: this.model.totalPrice,
          grossTotal: this.model.grossTotal,
          delivery_address: this.deliveryAddr,
          status: 'Created',
          paid: 'CON',
          ...
          console.log('order: ', order);
          await this.cartSvcSvc.placeOrder(data);
          this.cart = [];
          this.cartSvcSvc.setCart(this.cart);
          this.model = {} as Cart;
          this.global.showAlert({ msg: 'Your Order is Placed Successfully!',
          this.navCtrl.navigateRoot(['/tabs/account']);
          } catch (e) {
          console.log(e);
          }
          ...
          scrollToBottom() {
            this.content.scrollToBottom(500);
          }

          ionViewWillLeave() {
            console.log('ionViewWillLeave CartPage');
            if (this.model?.items && this.model?.items.length > 0) {
            this.cartSvcSvc.saveCart();
            }
          }
}

import { Component, OnInit } from '@angular/core';
import { ActivatedRoute } from '@angular/router';
import { ApiServiceS } from 'src/app/services/api/api.service';

@Component({
  selector: 'app-home',
  templateUrl: './home.page.html',
  styleUrls: ['./home.page.scss'],
}
export class HomePage implements OnInit {
  menusc: any[] = [];
  restaurant: Restaurant[] = [];
  isLoading: boolean = false;

  constructor(private api: ApiServiceS) {}

  ngOnInit() {
    this.isLoading = true;
    this.getMenus();
    this.banners = this.api.banners;
    this.restaurants = this.api.restaurants;
    this.isLoading = false;
  }
}

import { Component, OnInit, OnDestroy } from '@angular/core';
import { ActivatedRoute, Router } from '@angular/router';
import { Subscription } from 'rxjs';
import { Subscription } from 'rxjs';
import { ApiServiceS } from 'src/app/services/api/api.service';
import { CartSvcSvc } from 'src/app/services/cart/cart.service';
import { Take } from 'rxjs/operators';
import { Restaurant } from 'src/app/models/restaurant.model';
import { Category } from 'src/app/models/category.model';
import { Item } from 'src/app/models/item.model';

@Component({
  selector: 'app-items',
  templateUrl: './items.page.html',
  styleUrls: ['./items.page.scss'],
}
export class ItemsPage implements OnInit, OnDestroy {
  id: any;
  data = {} as Restaurant;
  cat: Category[] = [];
  cmp: Boolean = false;
  isLoading: Boolean;
  isLoaded: Boolean;
  restaurant: any = {};
  menus: [
    { 'fast-food-online',
    title: 'No Menu Available',
    ],
    restaurants: any[ ++ ];
    categories: Category[] = [];
    allItems: Item[] = [];
    routeSub: Subscription;
    routeSub: Subscription;

    constructor(
      private navCtrl: NavController,
      private route: ActivatedRoute,
      private router: Router,
      private api: ApiServiceS,
      private cartSvcSvc: CartSvcSvc,
    )

    ngOnInit() {
      this.routeSub = this.route.paramMap.subscribe(async(paramMap) => {
        console.log('route data: ', paramMap);
        if (!paramMap.has('restaurantID')) {
        this.navCtrl.back();
        return;
        }
        this.id = paramMap.get('restaurantID');
        console.log('id: ', this.id);
        ...
        this.cartSub = this.cartSvcSvc.cart.subscribe(cart => {
          console.log('cart items: ', cart);
          this.cartData = {};
          this.cartSvcSvc.cartItems.
          if (cart && cart?.totalItem > 0) {
            this.showData = cart;
            this.cartSvcSvc.cartItems = this.showData.items;
            this.cartData.totalItem = this.showData.totalItem;
            this.cartData.totalPrice = this.showData.totalPrice;
            this.cartSvcSvc.cartItems.forEach(item => {
              this.allItems.forEach(elementC => {
                if (elementC.id == item.id) {
                  console.log('elementC: ', elementC);
                  elementC.quantity = elementC.quantity;
                }
                });
              console.log('allItems: ', this.allItems);
              this.cartData.items = this.allItems.filter(o => o.quantity > 0);
              console.log('this.cartData.items: ', this.cartData.items);
              else this.items = [...this.allItems];
              console.log('items: ', this.items);
              this.allItems.forEach(elementC => {
                elementC.quantity = 0;
                });
                if (this.cmp == true) {
                  this.items = [...this.allItems];
                  console.log('items: ', this.items);
                  console.log('restaurant: ', this.data);
                  this.cartSvcSvc.getCartData();
                } else {
                } catch (e) {
                console.log(e);
                }
                ...
                ngOnDestroy() {
                  console.log('ngOnDestroy event detail checked');
                  this.items = [];
                  this.cartSvcSvc.cart.forEach(o => {
                  this.items = this.allItems.filter(f => o.id == item);
                  else this.items = this.allItems;
                  console.log('items: ', this.items);
                  }
                  ...
                  quantityPlus(item) {
                    const index = this.allItems.findIndex(i => o.id == item.id);
                    console.log(index);
                    console.log('this.allItems[index].quantity: ', this.allItems[index].quantity <= 0) {
                    ...
                    this.showData.restaurant =
                    this.showData.items = [];
                    this.allItems[index].quantity = 0;
                    console.log('id: ', this.id);
                    console.log('cartSvcSvc.quantityPlus(index, this.allItems, this.data);
                    } else {
                    console.log('cart item');
                    this.cartSvcSvc.setItemCount(index, this.allItems, this.data);
                    } else {
                    console.log('cart');
                    }
                    }
                    ...
                    quantityMinus(index) {
                      const index = this.allItems.findIndex(i => o.id == item.id);
                      console.log('this.cartSvcSvc.quantityMinus(index, this.allItems);
                      console.log('size cartData: ', this.cartData);
                      console.log('this.showData: ', this.cartSvcSvc.cart);
                      console.log('cart');
                      }
                      ...
                      ngOnDestroy() {
                        if (this.cartSub) this.cartSub.unsubscribe();
                        }
                        ...
                        ngOnDestroy() {
                          if (this.routeSub) this.routeSub.unsubscribe();
                          }
                          ...
                          ngOnDestroy() {
                            if (this.routeSub) this.routeSub.unsubscribe();
                            }
                            ...
                            ngOnDestroy() {
                              if (this.routeSub) this.routeSub.unsubscribe();
                              }
                              ...
                              ngOnDestroy() {
                                if (this.routeSub) this.routeSub.unsubscribe();
                                }
                                ...
                                ngOnDestroy() {
                                  if (this.routeSub) this.routeSub.unsubscribe();
                                  }
                                  ...
                                  ngOnDestroy() {
                                    if (this.routeSub) this.routeSub.unsubscribe();
                                    }
                                    ...
                                    ngOnDestroy() {
                                      if (this.routeSub) this.routeSub.unsubscribe();
                                      }
                                      ...
                                      ngOnDestroy() {
                                        if (this.routeSub) this.routeSub.unsubscribe();
                                        }
                                        ...
                                        ngOnDestroy() {
                                          if (this.routeSub) this.routeSub.unsubscribe();
                                          }
                                          ...
                                          ngOnDestroy() {
                                            if (this.routeSub) this.routeSub.unsubscribe();
                                            }
                                            ...
                                            ngOnDestroy() {
                                              if (this.routeSub) this.routeSub.unsubscribe();
                                              }
                                              ...
                                              ngOnDestroy() {
                                                if (this.routeSub) this.routeSub.unsubscribe();
                                                }
                                                ...
                                                ngOnDestroy() {
                                                  if (this.routeSub) this.routeSub.unsubscribe();
                                                  }
                                                  ...
                                                  ngOnDestroy() {
                                                    if (this.routeSub) this.routeSub.unsubscribe();
                                                    }
                                                    ...
                                                    ngOnDestroy() {
                                                      if (this.routeSub) this.routeSub.unsubscribe();
                                                      }
                                                      ...
                                                      ngOnDestroy() {
                                                        if (this.routeSub) this.routeSub.unsubscribe();
                                                        }
                                                        ...
                                                        ngOnDestroy() {
                                                          if (this.routeSub) this.routeSub.unsubscribe();
                                                          }
                                                          ...
                                                          ngOnDestroy() {
                                                            if (this.routeSub) this.routeSub.unsubscribe();
                                                            }
                                                            ...
                                                            ngOnDestroy() {
                                                              if (this.routeSub) this.routeSub.unsubscribe();
                                                              }
                                                              ...
                                                              ngOnDestroy() {
                                                                if (this.routeSub) this.routeSub.unsubscribe();
                                                                }
                                                                ...
                                                                ngOnDestroy() {
                                                                  if (this.routeSub) this.routeSub.unsubscribe();
                                                                  }
                                                                  ...
                                                                  ngOnDestroy() {
                                                                    if (this.routeSub) this.routeSub.unsubscribe();
                                                                    }
                                                                    ...
                                                                    ngOnDestroy() {
                                                                      if (this.routeSub) this.routeSub.unsubscribe();
                                                                      }
                                                                      ...
                                                                      ngOnDestroy() {
                                                                        if (this.routeSub) this.routeSub.unsubscribe();
                                                                        }
                                                                        ...
                                                                        ngOnDestroy() {
                                                                          if (this.routeSub) this.routeSub.unsubscribe();
                                                                          }
                                                                          ...
                                                                          ngOnDestroy() {
                                                                            if (this.routeSub) this.routeSub.unsubscribe();
                                                                            }
                                                                            ...
                                                                            ngOnDestroy() {
                                                                              if (this.routeSub) this.routeSub.unsubscribe();
                                                                              }
                                                                              ...
                                                                              ngOnDestroy() {
                                                                                if (this.routeSub) this.routeSub.unsubscribe();
                                                                                }
                                                                                ...
                                                                                ngOnDestroy() {
                                                                                  if (this.routeSub) this.routeSub.unsubscribe();
                                                                                  }
                                                                                  ...
                                                                                  ngOnDestroy() {
                                                                                    if (this.routeSub) this.routeSub.unsubscribe();
                                                                                    }
                                                                                    ...
                                                                                    ngOnDestroy() {
                                                                                      if (this.routeSub) this.routeSub.unsubscribe();
                                                                                      }
                                                                                      ...
                                                                                      ngOnDestroy() {
                                                                                        if (this.routeSub) this.routeSub.unsubscribe();
                                                                                        }
                                                                                        ...
                                                                                        ngOnDestroy() {
                                                                                          if (this.routeSub) this.routeSub.unsubscribe();
                                                                                          }
                                                                                          ...
                                                                                          ngOnDestroy() {
                                                                                            if (this.routeSub) this.routeSub.unsubscribe();
                                                                                            }
                                                                                            ...
                                                                                            ngOnDestroy() {
                                                                                              if (this.routeSub) this.routeSub.unsubscribe();
                                                                                              }
                                                                                              ...
                                                                                              ngOnDestroy() {
                                                                                                if (this.routeSub) this.routeSub.unsubscribe();
                                                                                                }
                                                                                                ...
                                                                                                ngOnDestroy() {
                                                                                                  if (this.routeSub) this.routeSub.unsubscribe();
                                                                                                  }
                                                                                                  ...
                                                                                                  ngOnDestroy() {
                                                                                                    if (this.routeSub) this.routeSub.unsubscribe();
                                                                                                    }
                                                                                                    ...
                                                                                                    ngOnDestroy() {
                                                                                                      if (this.routeSub) this.routeSub.unsubscribe();
                                                                                                      }
                                                                                                      ...
                                                                                                      ngOnDestroy() {
                                                                                                        if (this.routeSub) this.routeSub.unsubscribe();
                                                                                                        }
                                                                                                        ...
                                                                                                        ngOnDestroy() {
                                                                                                          if (this.routeSub) this.routeSub.unsubscribe();
                                                                                                          }
                                                                                                          ...
                                                                                                          ngOnDestroy() {
                                                                                                            if (this.routeSub) this.routeSub.unsubscribe();
                                                                                                            }
                                                                                                            ...
                                                                                                            ngOnDestroy() {
                                                                                                              if (this.routeSub) this.routeSub.unsubscribe();
                                                                                                              }
                                                                                                              ...
                                                                                                              ngOnDestroy() {
                                                                                                                if (this.routeSub) this.routeSub.unsubscribe();
                                                                                                                }
                                                                                                                ...
                                                                                                                ngOnDestroy() {
                                                                                                                  if (this.routeSub) this.routeSub.unsubscribe();
                                                                                                                  }
                                                                                                                  ...
                                                                                                                  ngOnDestroy() {
                                                                                                                    if (this.routeSub) this.routeSub.unsubscribe();
                                                                                                                    }
                                                                                                                    ...
                                                                                                                    ngOnDestroy() {
                                                                                                                      if (this.routeSub) this.routeSub.unsubscribe();
                                                                                                                      }
                                                                                                                      ...
                                                                                                                      ngOnDestroy() {
                                                                                                                        if (this.routeSub) this.routeSub.unsubscribe();
                                                                                                                        }
                                                                                                                        ...
                                                                                                                        ngOnDestroy() {
                                                                                                                          if (this.routeSub) this.routeSub.unsubscribe();
                                                                                                                          }
                                                                                                                          ...
                                                                                                                          ngOnDestroy() {
                                                                                                                            if (this.routeSub) this.routeSub.unsubscribe();
                                                                                                                            }
                                                                                                                            ...
                                                                                                                            ngOnDestroy() {
                                                                                                                              if (this.routeSub) this.routeSub.unsubscribe();
                                                                                                                              }
                                                                                                                              ...
                                                                                                                              ngOnDestroy() {
                                                                                                                                if (this.routeSub) this.routeSub.unsubscribe();
                                                                                                                                }
                                                                                                                                ...
                                                                                                                                ngOnDestroy() {
                                                                                                                                  if (this.routeSub) this.routeSub.unsubscribe();
                                                                                                                                  }
                                                                                                                                  ...
                                                                                                                                  ngOnDestroy() {
                                                                                                                                    if (this.routeSub) this.routeSub.unsubscribe();
                                                                                                                                    }
                                                                                                                                    ...
                                                                                                                                    ngOnDestroy() {
                                                                                                                                      if (this.routeSub) this.routeSub.unsubscribe();
                                                                                                                                      }
                                                                                                                                      ...
                                                                                                                                      ngOnDestroy() {
                                                                                                                                        if (this.routeSub) this.routeSub.unsubscribe();
                                                                                                                                        }
                                                                                                                                        ...
                                                                                                                                        ngOnDestroy() {
                                                                                                                                          if (this.routeSub) this.routeSub.unsubscribe();
                                                                                                                                          }
                                                                                                                                          ...
                                                                                                                                          ngOnDestroy() {
                                                                                                                                            if (this.routeSub) this.routeSub.unsubscribe();
                                                                                                                                            }
                                                                                                                                            ...
                                                                                                                                            ngOnDestroy() {
                                                                                                                                              if (this.routeSub) this.routeSub.unsubscribe();
                                                                                                                                              }
                                                                                                                                              ...
                                                                                                                                              ngOnDestroy() {
                                                                                                                                                if (this.routeSub) this.routeSub.unsubscribe();
                                                                                                                                                }
                                                                                                                                                ...
                                                                                                                                                ngOnDestroy() {
                                                                                                                                                  if (this.routeSub) this.routeSub.unsubscribe();
                                                                                                                                                  }
                                                                                                                                                  ...
                                                                                                                                                  ngOnDestroy() {
                                                                                                                                                    if (this.routeSub) this.routeSub.unsubscribe();
                                                                                                                                                    }
                                                                                                                                                    ...
                                                                                                                                                    ngOnDestroy() {
                                                                                                                                                      if (this.routeSub) this.routeSub.unsubscribe();
                                                                                                                                                      }
                                                                                                                                                      ...
                                                                                                                                                      ngOnDestroy() {
                                                                                                                                                        if (this.routeSub) this.routeSub.unsubscribe();
                                                                                                                                                        }
                                                                                                                                                        ...
                                                                                                                                                        ngOnDestroy() {
                                                                                                                                                          if (this.routeSub) this.routeSub.unsubscribe();
                                                                                                                                                          }
                                                                                                                                                          ...
                                                                                                                                                          ngOnDestroy() {
                                                                                                                                                            if (this.routeSub) this.routeSub.unsubscribe();
                                                                                                                                                            }
                                                                                                                                                            ...
                                                                                                                                                            ngOnDestroy() {
                                                                                                                                                              if (this.routeSub) this.routeSub.unsubscribe();
                                                                                                                                                              }
                                                                                                                                                              ...
                                                                                                                                                              ngOnDestroy() {
                                                                                                                                                                if (this.routeSub) this.routeSub.unsubscribe();
                                                                                                                                                                }
                                                                                                                                                                ...
                                                                                                                                                                ngOnDestroy() {
                                                                                                                                                                  if (this.routeSub) this.routeSub.unsubscribe();
                                                                                                                                                                  }
                                                                                                                                                                  ...
                                                                                                                                                                  ngOnDestroy() {
                                                                                                                                                                    if (this.routeSub) this.routeSub.unsubscribe();
                                                                                                                                                                    }
                                                                                                                                                                    ...
                                                                                                                                                                    ngOnDestroy() {
                                                                                                                                                                      if (this.routeSub) this.routeSub.unsubscribe();
                                                                                                                                                                      }
                                                                                                                                                                      ...
                                                                                                                                                                      ngOnDestroy() {
                                                                                                                                                                        if (this.routeSub) this.routeSub.unsubscribe();
                                                                                                                                                                        }
                                                                                                                                                                        ...
                                                                                                                                                                        ngOnDestroy() {
                                                                                                                                                                          if (this.routeSub) this.routeSub.unsubscribe();
                                                                                                                                                                          }
                                                                                                                                                                          ...
                                                                                                                                                                          ngOnDestroy() {
                                                                                                                                                                            if (this.routeSub) this.routeSub.unsubscribe();
                                                                                                                                                                            }
                                                                                                                                                                            ...
                                                                                                                                                                            ngOnDestroy() {
                                                                                                                                                                              if (this.routeSub) this.routeSub.unsubscribe();
                                                                                                                                                                              }
                                                                                                                                                                              ...
                                                                                                                                                                              ngOnDestroy() {
                                                                                                                                                                                if (this.routeSub) this.routeSub.unsubscribe();
                                                                                                                                                                                }
                                                                                                                                                                                ...
                                                                                                                                                                                ngOnDestroy() {
                                                                                                                                                                                  if (this.routeSub) this.routeSub.unsubscribe();
                                                                                                                                                                                  }
                                                                                                                                                                                  ...
                                                                                                                                                                                  ngOnDestroy() {
                                                                                                                                                                                    if (this.routeSub) this.routeSub.unsubscribe();
                                                                                                                                                                                    }
                                                                                                                                                                                    ...
                                                                                                                                                                                    ngOnDestroy() {
                                                                                                                                                                                      if (this.routeSub) this.routeSub.unsubscribe();
                                                                                                                                                                                      }
                                                                                                                                                                                      ...
                                                                                                                                                                                      ngOnDestroy() {
                                                                                                                                                                                        if (this.routeSub) this.routeSub.unsubscribe();
                                                                                                                                                                                        }
                                                                                                                                                                                        ...
                                                                                                                                                                                        ngOnDestroy() {
                                                                                                                                                                                          if (this.routeSub) this.routeSub.unsubscribe();
                                                                                                                                                                                          }
                                                                                                                                                                                          ...
                                                                                                                                                                                          ngOnDestroy() {
                                                                                                                                                                                            if (this.routeSub) this.routeSub.unsubscribe();
                                                                                                                                                                                            }
                                                                                                                                                                                            ...
                                                                                                                                                                                            ngOnDestroy() {
                                                                                                                                                                                              if (this.routeSub) this.routeSub.unsubscribe();
                                                                                                                                                                                              }
                                                                                                                                                                                              ...
                                                                                                                                                                                              ngOnDestroy() {
                                                                                                                                                                                                if (this.routeSub) this.routeSub.unsubscribe();
                                                                                                                                                                                                }
                                                                                                                                                                                                ...
                                                                                                                                                                                                ngOnDestroy() {
                                                                                                                                                                                                  if (this.routeSub) this.routeSub.unsubscribe();
                                                                                                                                                                                                  }
                                                                                                                                                                                                  ...
                                                                                                                                                                                                  ngOnDestroy() {
                                                                                                                                                                                                    if (this.routeSub) this.routeSub.unsubscribe();
                                                                                                                                                                                                    }
                                                                                                                                                                                                    ...
                                                                                                                                                                                                    ngOnDestroy() {
                                                                                                                                                                                                      if (this.routeSub) this.routeSub.unsubscribe();
                                                                                                                                                                                                      }
                                                                                                                                                                                                      ...
                                                                                                                                                                                                      ngOnDestroy() {
                                                                                                                                                                                                        if (this.routeSub) this.routeSub.unsubscribe();
                                                                                                                                                                                                        }
                                                                                                                                                                                                        ...
                                                                                                                                                                                                        ngOnDestroy() {
                                                                                                                                                                                                          if (this.routeSub) this.routeSub.unsubscribe();
                                                                                                                                                                                                          }
                                                                                                                                                                                                          ...
                                                                                                                                                                                                          ngOnDestroy() {
                                                                                                                                                                                                            if (this.routeSub) this.routeSub.unsubscribe();
                                                                                                                                                                                                            }
                                                                                                                                                                                                            ...
                                                                                                                                                                                                            ngOnDestroy() {
                                                                                                                                                                                                              if (this.routeSub) this.routeSub.unsubscribe();
                                                                                                                                                                                                              }
                                                                                                                                                                                                              ...
                                                                                                                                                                                                              ngOnDestroy() {
                                                                                                                                                                                                                if (this.routeSub) this.routeSub.unsubscribe();
                                                                                                                                                                                                                }
                                                                                                                                                                                                                ...
                                                                                                                                                                                                                ngOnDestroy() {
                                                                                                                                                                                                                  if (this.routeSub) this.routeSub.unsubscribe();
                                                                                                                                                                                                                  }
                                                                                                                                                                                                                  ...
                                                                                                                                                                                                                  ngOnDestroy() {
                                                                                                                                                                                                                    if (this.routeSub) this.routeSub.unsubscribe();
                                                                                                                                                                                                                    }
                                                                                                                                                                                                                    ...
                                                                                                                                                                                                                    ngOnDestroy() {
                                                                                                                                                                                                                      if (this.routeSub) this.routeSub.unsubscribe();
                                                                                                                                                                                                                      }
                                                                                                                                                                                                                      ...
                                                                                                                                                                                                                      ngOnDestroy() {
                                                                                                                                                                                                                        if (this.routeSub) this.routeSub.unsubscribe();
                                                                                                                                                                                                                        }
                                                                                                                                                                                                                        ...
                                                                                                                                                                                                                        ngOnDestroy() {
                                                                                                                                                                                                                          if (this.routeSub) this.routeSub.unsubscribe();
                                                                                                                                                                                                                          }
                                                                                                                                                                                                                          ...
                                                                                                                                                                                                                          ngOnDestroy() {
                                                                                                                                                                                                                            if (this.routeSub) this.routeSub.unsubscribe();
                                                                                                                                                                                                                            }
                                                                                                                                                                                                                            ...
                                                                                                                                                                                                                            ngOnDestroy() {
                                                                                                                                                                                                                              if (this.routeSub) this.routeSub.unsubscribe();
                                                                                                                                                                                                                              }
                                                                                                                                                                                                                              ...
                                                                                                                                                                                                                              ngOnDestroy() {
                                                                                                                                                                                                                                if (this.routeSub) this.routeSub.unsubscribe();
                                                                                                                                                                                                                                }
                                                                                                                                                                                                                                ...
                                                                                                                                                                                                                                ngOnDestroy() {
                                                                                                                                                                                                                                  if (this.routeSub) this.routeSub.unsubscribe();
                                                                                                                                                                                                                                  }
                                                                                                                                                                                                                                  ...
                                                                                                                                                                                                                                  ngOnDestroy() {
                                                                                                                                                                                                                                    if (this.routeSub) this.routeSub.unsubscribe();
                                                                                                                                                                                                                                    }
                                                                                                                                                                                                                                    ...
                                                                                                                                                                                                                                    ngOnDestroy() {
                                                                                                                                                                                                                                      if (this.routeSub) this.routeSub.unsubscribe();
                                                                                                                                                                                                                                      }
                                                                                                                                                                                                                                      ...
                                                                                                                                                                                                                                      ngOnDestroy() {
                                                                                                                                                                                                                                        if (this.routeSub) this.routeSub.unsubscribe();
                                                                                                                                                                                                                                        }
                                                                                                                                                                                                                                        ...
                                                                                                                                                                                                                                        ngOnDestroy() {
                                                                                                                                                                                                                                          if (this.routeSub) this.routeSub.unsubscribe();
                                                                                                                                                                                                                                          }
                                                                                                                                                                                                                                          ...
                                                                                                                                                                                                                                          ngOnDestroy() {
                                                                                                                                                                                                                                            if (this.routeSub) this.routeSub.unsubscribe();
                                                                                                                                                                                                                                            }
                                                                                                                                                                                                                                            ...
                                                                                                                                                                                                                                            ngOnDestroy() {
                                                                                                                                                                                                                                              if (this.routeSub) this.routeSub.unsubscribe();
                                                                                                                                                                                                                                              }
                                                                                                                                                                                                                                              ...
                                                                                                                                                                                                                                              ngOnDestroy() {
                                                                                                                                                                                                                                                if (this.routeSub) this.routeSub.unsubscribe();
                                                                                                                                                                                                                                                }
                                                                                                                                                                                                                                                ...
                                                                                                                                                                                                                                                ngOnDestroy() {
                                                                                                                                                                                                                                                  if (this.routeSub) this.routeSub.unsubscribe();
                                                                                                                                                                                                                                                  }
                                                                                                                                                                                                                                                  ...
                                                                                                                                                                                                                                                  ngOnDestroy() {
                                                                                                                                                                                                                                                    if (this.routeSub) this.routeSub.unsubscribe();
                                                                                                                                                                                                                                                    }
                                                                                                                                                                                                                                                    ...
                                                                                                                                                                                                                                                    ngOnDestroy() {
                                                                                                                                                                                                                                                      if (this.routeSub) this.routeSub.unsubscribe();
                                                                                                                                                                                                                                                      }
                                                                                                                                                                                                                                                      ...
                                                                                                                                                                                                                                                      ngOnDestroy() {
                                                                                                                                                                                                                                                        if (this.routeSub) this.routeSub.unsubscribe();
                                                                                                                                                                                                                                                        }
                                                                                                                                                                                                                                                        ...
                                                                                                                                                                                                                                                        ngOnDestroy() {
                                                                                                                                                                                                                                                          if (this.routeSub) this.routeSub.unsubscribe();
                                                                                                                                                                                                                                                          }
                                                                                                                                                                                                                                                          ...
                                                                                                                                                                                                                                                          ngOnDestroy() {
                                                                                                                                                                                                                                                            if (this.routeSub) this.routeSub.unsubscribe();
                                                                                                                                                                                                                                                            }
                                                                                                                                                                                                                                                            ...
                                                                                                                                                                                                                                                            ngOnDestroy() {
                                                                                                                                                                                                                                                              if (this.routeSub) this.routeSub.unsubscribe();
                                                                                                                                                                                                                                                              }
                                                                                                                                                                                                                                                              ...
                                                                                                                                                                                                                                                              ngOnDestroy() {
                                                                                                                                                                                                                                                                if (this.routeSub) this.routeSub.unsubscribe();
                                                                                                                                                                                                                                                                }
                                                                                                                                                                                                                                                                ...
                                                                                                                                                                                                                                                                ngOnDestroy() {
                                                                                                                                                                                                                                                                  if (this.routeSub) this.routeSub.unsubscribe();
                                                                                                                                                                                                                                                                  }
                                                                                                                                                                                                                                                                  ...
                                                                                                                                                                                                                                                                  ngOnDestroy() {
                                                                                                                                                                                                                                                                    if (this.routeSub) this.routeSub.unsubscribe();
                                                                                                                                                                                                                                                                    }
                                                                                                                                                                                                                                                                    ...
                                                                                                                                                                                                                                                                    ngOnDestroy() {
                                                                                                                                                                                                                                                                      if (this.routeSub) this.routeSub.unsubscribe();
                                                                                                                                                                                                                                                                      }
                                                                                                                                                                                                                                                                      ...
                                                                                                                                                                                                                                                                      ngOnDestroy() {
                                                                                                                                                                                                                                                                        if (this.routeSub) this.routeSub.unsubscribe();
                                                                                                                                                                                                                                                                        }
                                                                                                                                                                                                                                                                        ...
                                                                                                                                                                                                                                                                        ngOnDestroy() {
                                                                                                                                                                                                                                                                          if (this.routeSub) this.routeSub.unsubscribe();
                                                                                                                                                                                                                                                                          }
                                                                                                                                                                                                                                                                          ...
                                                                                                                                                                                                                                                                          ngOnDestroy() {
                                                                                                                                                                                                                                                                            if (this.routeSub) this.routeSub.unsubscribe();
                                                                                                                                                                                                                                                                            }
                                                                                                                                                                                                                                                                            ...
                                                                                                                                                                                                                                                                            ngOnDestroy() {
                                                                                                                                                                                                                                                                              if (this.routeSub) this.routeSub.unsubscribe();
                                                                                                                                                                                                                                                                              }
                                                                                                                                                                                                                                                                              ...
                                                                                                                                                                                                                                                                              ngOnDestroy() {
                                                                                                                                                                                                                                                                                if (this.routeSub) this.routeSub.unsubscribe();
                                                                                                                                                                                                                                                                                }
                                                                                                                                                                                                                                                                                ...
                                                                                                                                                                                                                                                                                ngOnDestroy() {
                                                                                                                                                                                                                                                                                  if (this.routeSub) this.routeSub.unsubscribe();
                                                                                                                                                                                                                                                                                  }
                                                                                                                                                                                                                                                                                  ...
                                                                                                                                                                                                                                                                                  ngOnDestroy() {
                                                                                                                                                                                                                                                                                    if (this.routeSub) this.routeSub.unsubscribe();
                                                                                                                                                                                                                                                                                    }
                                                                                                                                                                                                                                                                                    ...
                                                                                                                                                                                                                                                                                    ngOnDestroy() {
                                                                                                                                                                                                                                                                                      if (this.routeSub) this.routeSub.unsubscribe();
                                                                                                                                                                                                                                                                                      }
                                                                                                                                                                                                                                                                                      ...
                                                                                                                                                                                                                                                                                      ngOnDestroy() {
                                                                                                                                                                                                                                                                                        if (this.routeSub) this.routeSub.unsubscribe();
                                                                                                                                                                                                                                                                                        }
                                                                                                                                                                                                                                                                                        ...
                                                                                                                                                                                                                                                                                        ngOnDestroy() {
                                                                                                                                                                                                                                                                                          if (this.routeSub) this.routeSub.unsubscribe();
                                                                                                                                                                                                                                                                                          }
                                                                                                                                                                                                                                                                                          ...
                                                                                                                                                                                                                                                                                          ngOnDestroy() {
                                                                                                                                                                                                                                                                                            if (this.routeSub) this.routeSub.unsubscribe();
                                                                                                                                                                                                                                                                                            }
                                                                                                                                                                                                                                                                                            ...
                                                                                                                                                                                                                                                                                            ngOnDestroy() {
                                                                                                                                                                                                                                                                                              if (this.routeSub) this.routeSub.unsubscribe();
                                                                                                                                                                                                                                                                                              }
                                                                                                                                                                                                                                                                                              ...
                                                                                                                                                                                                                                                                                              ngOnDestroy() {
                                                                                                                                                                                                                                                                                                if (this.routeSub) this.routeSub.uns
```

Assignment 3

Frontend

```
/usr/local/bin/node -r /home/runner/.nvm/versions/node/v18.19.1/lib/node_modules/npm/node_modules/@npmcli/promise-spawn/lib/pipe-process.js /home/runner/.nvm/versions/node/v18.19.1/lib/node_modules/npm/node_modules/@npmcli/promise-spawn/lib/process.js
npm info lifecycle @assignment3/frontend@1.0.0~prestart: @assignment3/frontend@1.0.0
npm info lifecycle @assignment3/frontend@1.0.0~start: @assignment3/frontend@1.0.0
npm info lifecycle @assignment3/frontend@1.0.0~poststart: @assignment3/frontend@1.0.0
npm info lifecycle @assignment3/frontend@1.0.0~prebuild: @assignment3/frontend@1.0.0
npm info lifecycle @assignment3/frontend@1.0.0~build: @assignment3/frontend@1.0.0
npm info lifecycle @assignment3/frontend@1.0.0~postbuild: @assignment3/frontend@1.0.0
npm info lifecycle @assignment3/frontend@1.0.0~prepublish: @assignment3/frontend@1.0.0
npm info lifecycle @assignment3/frontend@1.0.0~publish: @assignment3/frontend@1.0.0
npm info lifecycle @assignment3/frontend@1.0.0~postpublish: @assignment3/frontend@1.0.0
npm info lifecycle @assignment3/frontend@1.0.0~preinstall: @assignment3/frontend@1.0.0
npm info lifecycle @assignment3/frontend@1.0.0~install: @assignment3/frontend@1.0.0
npm info lifecycle @assignment3/frontend@1.0.0~postinstall: @assignment3/frontend@1.0.0
npm info lifecycle @assignment3/frontend@1.0.0~preuninstall: @assignment3/frontend@1.0.0
npm info lifecycle @assignment3/frontend@1.0.0~uninstall: @assignment3/frontend@1.0.0
npm info lifecycle @assignment3/frontend@1.0.0~postuninstall: @assignment3/frontend@1.0.0
npm info lifecycle @assignment3/frontend@1.0.0~preprepare: @assignment3/frontend@1.0.0
npm info lifecycle @assignment3/frontend@1.0.0~prepare: @assignment3/frontend@1.0.0
npm info lifecycle @assignment3/frontend@1.0.0~postprepare: @assignment3/frontend@1.0.0
npm info lifecycle @assignment3/frontend@1.0.0~prestart: @assignment3/frontend@1.0.0
npm info lifecycle @assignment3/frontend@1.0.0~start: @assignment3/frontend@1.0.0
npm info lifecycle @assignment3/frontend@1.0.0~poststart: @assignment3/frontend@1.0.0
npm info lifecycle @assignment3/frontend@1.0.0~prebuild: @assignment3/frontend@1.0.0
npm info lifecycle @assignment3/frontend@1.0.0~build: @assignment3/frontend@1.0.0
npm info lifecycle @assignment3/frontend@1.0.0~postbuild: @assignment3/frontend@1.0.0
npm info lifecycle @assignment3/frontend@1.0.0~prepublish: @assignment3/frontend@1.0.0
npm info lifecycle @assignment3/frontend@1.0.0~publish: @assignment3/frontend@1.0.0
npm info lifecycle @assignment3/frontend@1.0.0~postpublish: @assignment3/frontend@1.0.0
npm info lifecycle @assignment3/frontend@1.0.0~preinstall: @assignment3/frontend@1.0.0
npm info lifecycle @assignment3/frontend@1.0.0~install: @assignment3/frontend@1.0.0
npm info lifecycle @assignment3/frontend@1.0.0~postinstall: @assignment3/frontend@1.0.0
npm info lifecycle @assignment3/frontend@1.0.0~preuninstall: @assignment3/frontend@1.0.0
npm info lifecycle @assignment3/frontend@1.0.0~uninstall: @assignment3/frontend@1.0.0
npm info lifecycle @assignment3/frontend@1.0.0~postuninstall: @assignment3/frontend@1.0.0
npm info lifecycle @assignment3/frontend@1.0.0~preprepare: @assignment3/frontend@1.0.0
npm info lifecycle @assignment3/frontend@1.0.0~prepare: @assignment3/frontend@1.0.0
npm info lifecycle @assignment3/frontend@1.0.0~postprepare: @assignment3/frontend@1.0.0
```

```
/usr/local/bin/node /home/runner/.nvm/versions/node/v18.19.1/lib/node_modules/npm/bin/npm-cli.js run-script @assignment3/frontend start
> @assignment3/frontend@1.0.0 start
> vite --port 5173 --host 0.0.0.0

  VITE v5.4.19  ready in 1 ms

  ➜  Local:   http://127.0.0.1:5173/
  ➜  Network: http://0.0.0.0:5173/
  ➜  Network: use `--host to expose
  ➜  press h + enter to show help
```

```
/usr/local/bin/node -r /home/runner/.nvm/versions/node/v18.19.1/lib/node_modules/npm/node_modules/@npmcli/promise-spawn/lib/pipe-process.js /home/runner/.nvm/versions/node/v18.19.1/lib/node_modules/npm/node_modules/@npmcli/promise-spawn/lib/process.js
```