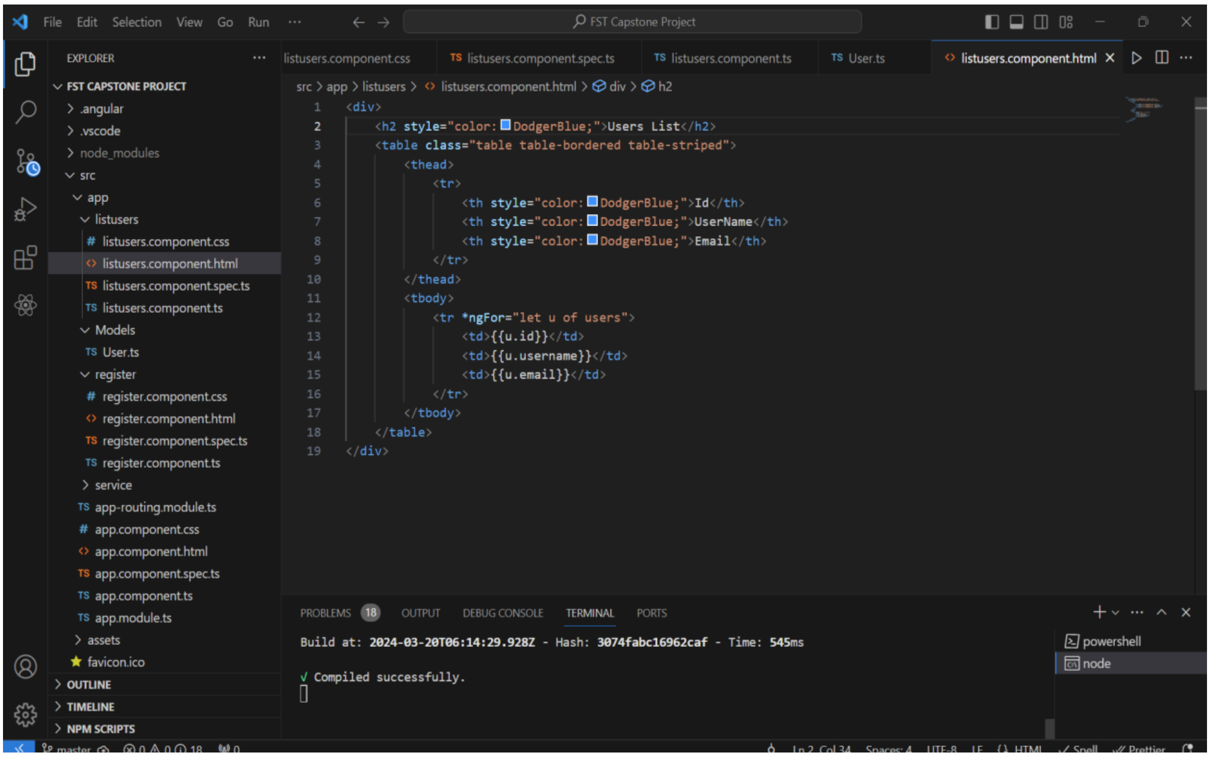
FSD-CAPSTONE PROJECT: Health Care Application.

Angular Part:-

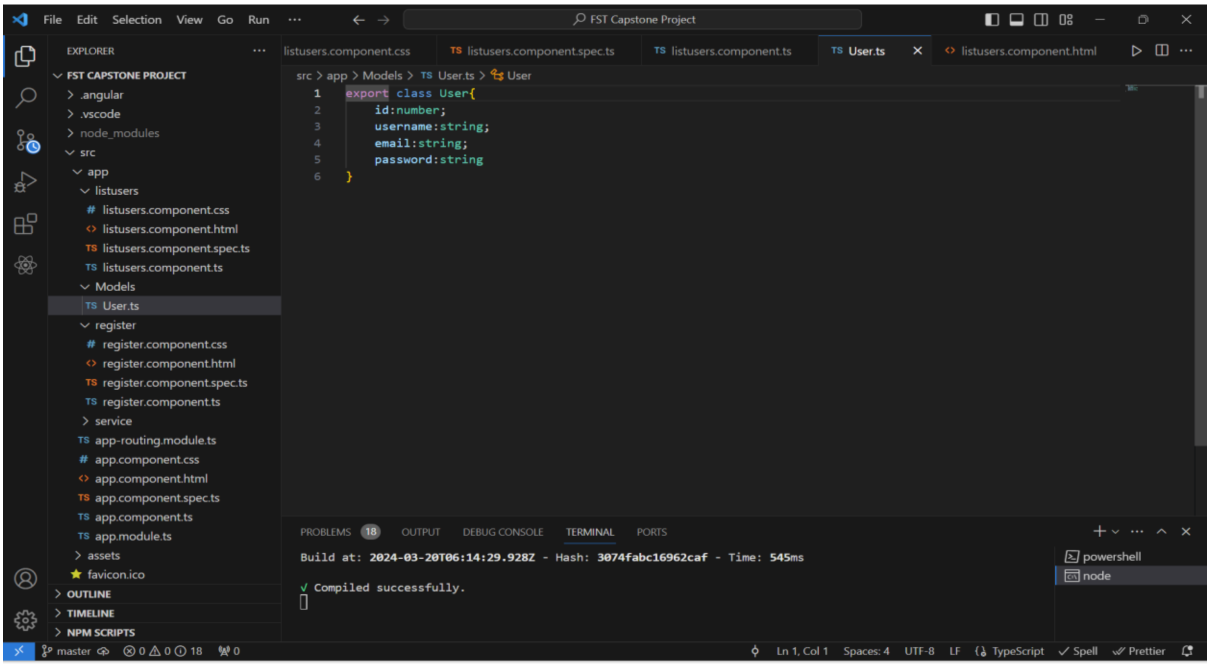
List users:



```html
<div>
    <h2 style="color:DodgerBlue;">Users List</h2>
    <table class="table table-bordered table-striped">
        <thead>
            <tr>
                <th style="color:DodgerBlue;">Id</th>
                <th style="color:DodgerBlue;">UserName</th>
                <th style="color:DodgerBlue;">Email</th>
            </tr>
        </thead>
        <tbody>
            <tr *ngFor="let u of users">
                <td>{{u.id}}</td>
                <td>{{u.username}}</td>
                <td>{{u.email}}</td>
            </tr>
        </tbody>
    </table>
</div>
```

```
Build at: 2024-03-20T06:14:29.928Z - Hash: 3074fabc16962caf - Time: 545ms

√ Compiled successfully.
```

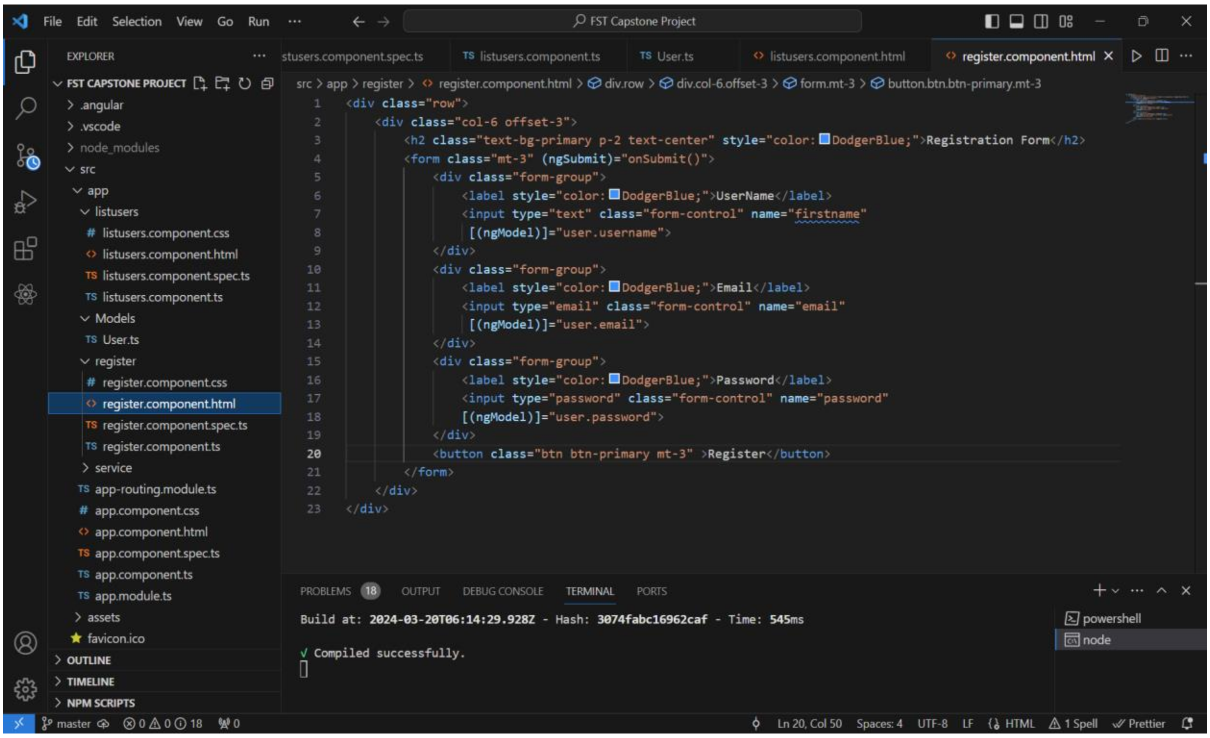
List User:



```typescript
export class User{
    id:number;
    username:string;
    email:string;
    password:string
}
```

```
Build at: 2024-03-20T06:14:29.928Z - Hash: 3074fabc16962caf - Time: 545ms

√ Compiled successfully.
```

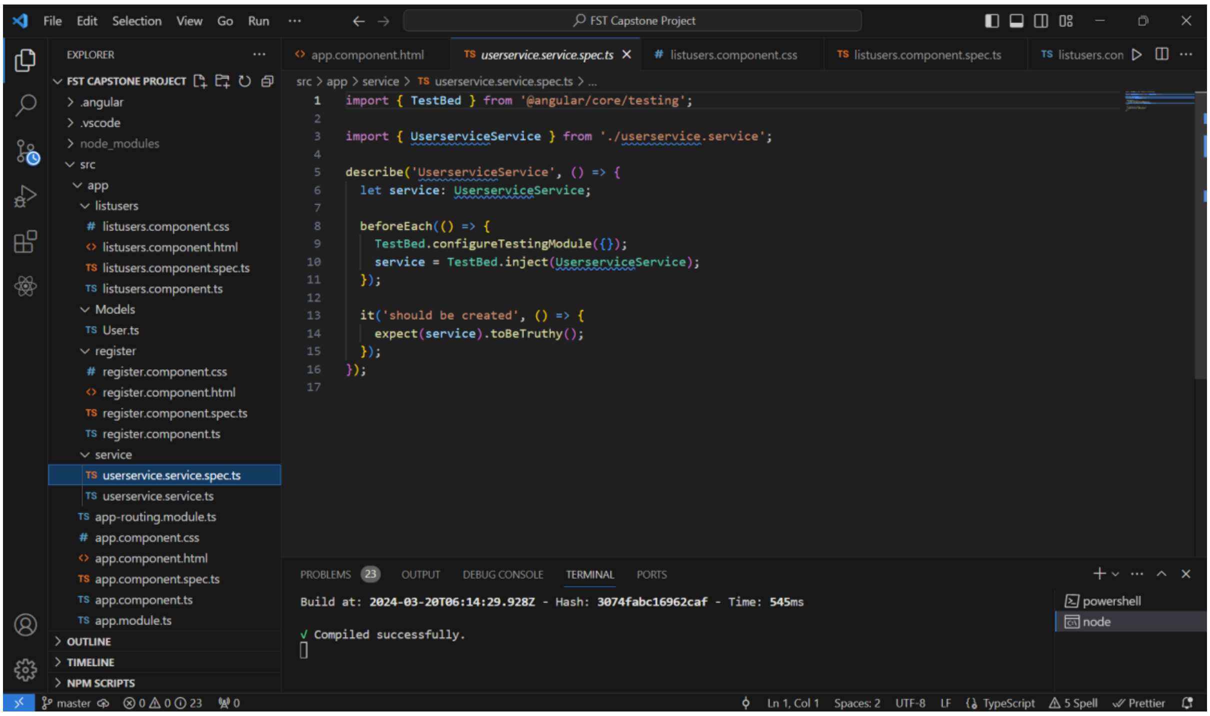
Register component:



```html
<div class="row">
    <div class="col-6 offset-3">
        <h2 class="text-bg-primary p-2 text-center" style="color:DodgerBlue;">Registration Form</h2>
        <form class="mt-3" (ngSubmit)="onSubmit()">
            <div class="form-group">
                <label style="color:DodgerBlue;">UserName</label>
                <input type="text" class="form-control" name="firstname"
                 [(ngModel)]="user.username">
            </div>
            <div class="form-group">
                <label style="color:DodgerBlue;">Email</label>
                <input type="email" class="form-control" name="email"
                 [(ngModel)]="user.email">
            </div>
            <div class="form-group">
                <label style="color:DodgerBlue;">Password</label>
                <input type="password" class="form-control" name="password"
                [(ngModel)]="user.password">
            </div>
            <button class="btn btn-primary mt-3" >Register</button>
        </form>
    </div>
</div>
```

```
Build at: 2024-03-20T06:14:29.928Z - Hash: 3074fabc16962caf - Time: 545ms

√ Compiled successfully.
```

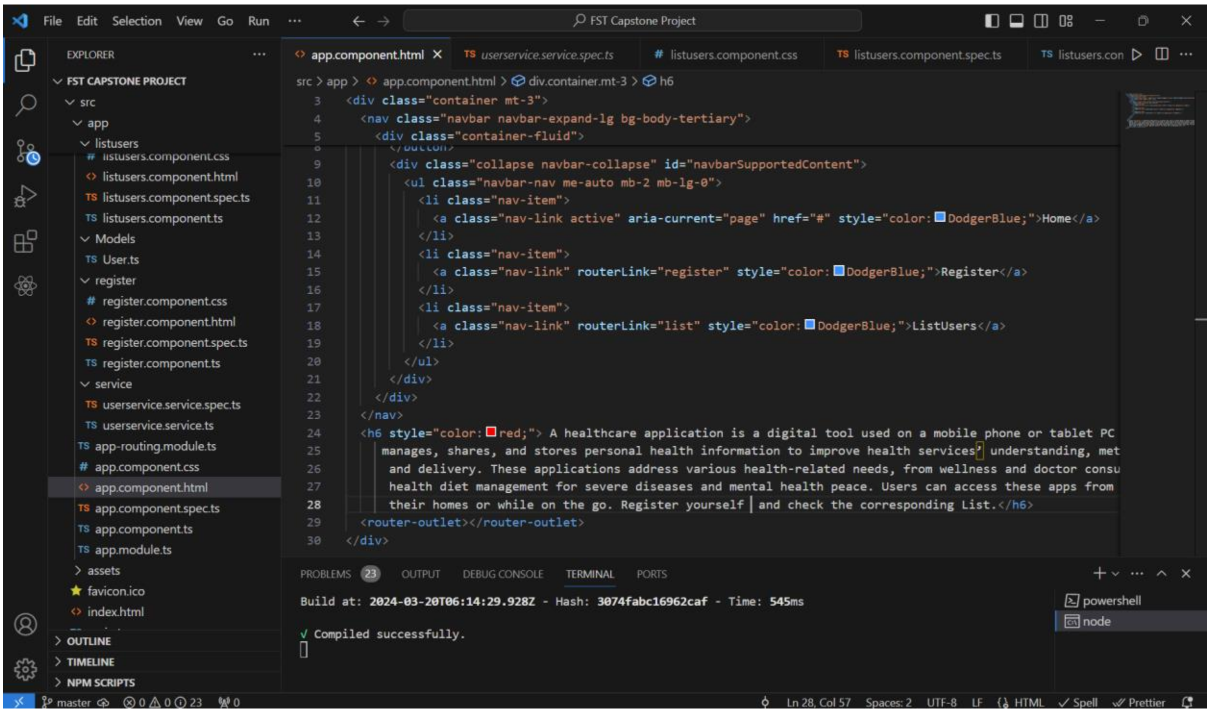
Service component:



```typescript
import { TestBed } from '@angular/core/testing';

import { UserserviceService } from './userservice.service';

describe('UserserviceService', () => {
  let service: UserserviceService;

  beforeEach(() => {
    TestBed.configureTestingModule({});
    service = TestBed.inject(UserserviceService);
  });

  it('should be created', () => {
    expect(service).toBeTruthy();
  });
});
```

```
Build at: 2024-03-20T06:14:29.928Z - Hash: 3074fabc16962caf - Time: 545ms

√ Compiled successfully.
```

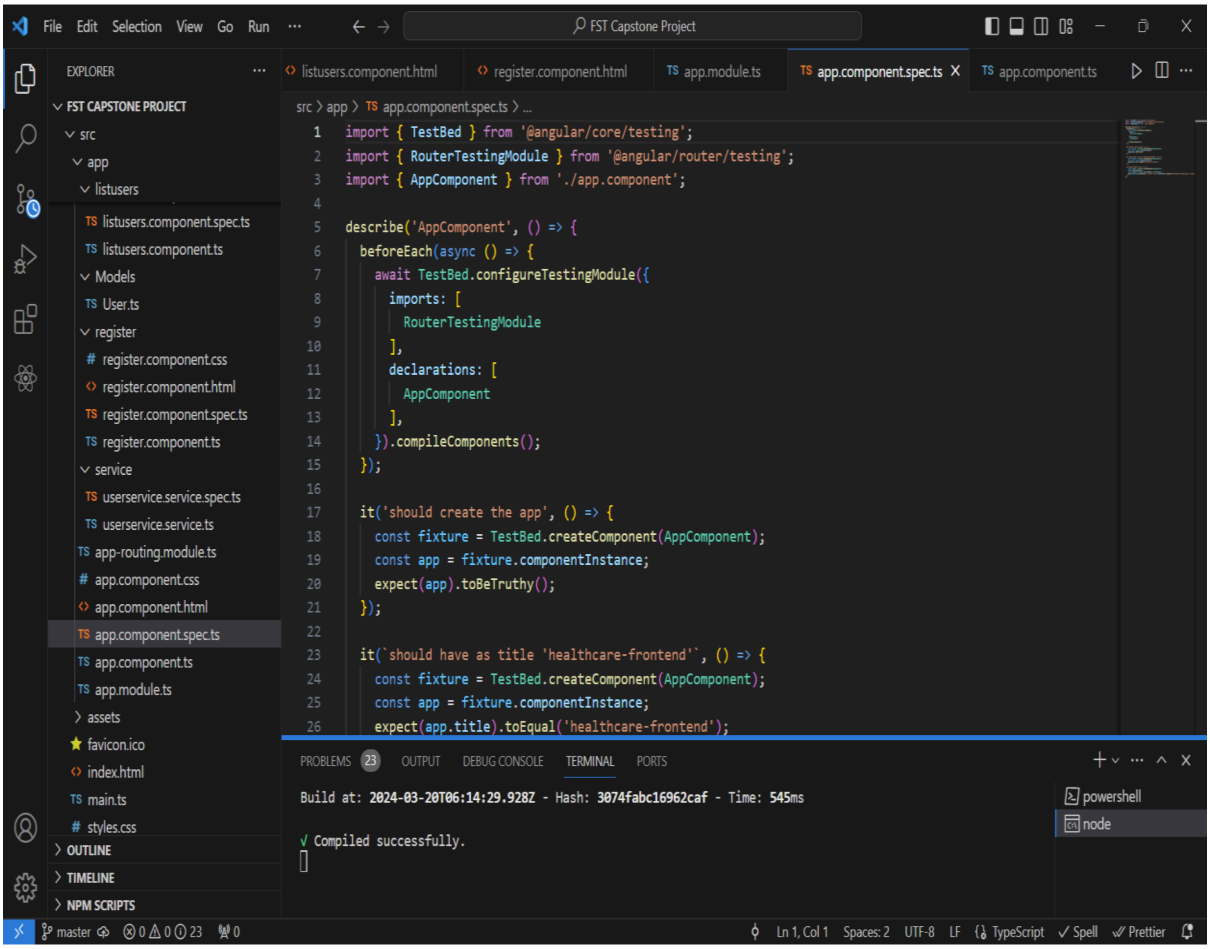
App Component:



```html
<div class="container mt-3">
  <nav class="navbar navbar-expand-lg bg-body-tertiary">
    <div class="container-fluid">
      </button>
      <div class="collapse navbar-collapse" id="navbarSupportedContent">
        <ul class="navbar-nav me-auto mb-2 mb-lg-0">
          <li class="nav-item">
            <a class="nav-link active" aria-current="page" href="#" style="color:DodgerBlue;">Home</a>
          </li>
          <li class="nav-item">
            <a class="nav-link" routerLink="register" style="color:DodgerBlue;">Register</a>
          </li>
          <li class="nav-item">
            <a class="nav-link" routerLink="list" style="color:DodgerBlue;">ListUsers</a>
          </li>
        </ul>
      </div>
    </div>
  </nav>
  <h6 style="color:red;"> A healthcare application is a digital tool used on a mobile phone or tablet PC
     manages, shares, and stores personal health information to improve health services' understanding, met
      and delivery. These applications address various health-related needs, from wellness and doctor consu
      health diet management for severe diseases and mental health peace. Users can access these apps from
      their homes or while on the go. Register yourself and check the corresponding List.</h6>
  <router-outlet></router-outlet>
</div>
```

```
Build at: 2024-03-20T06:14:29.928Z - Hash: 3074fabc16962caf - Time: 545ms

√ Compiled successfully.
```

App Component.spec.ts

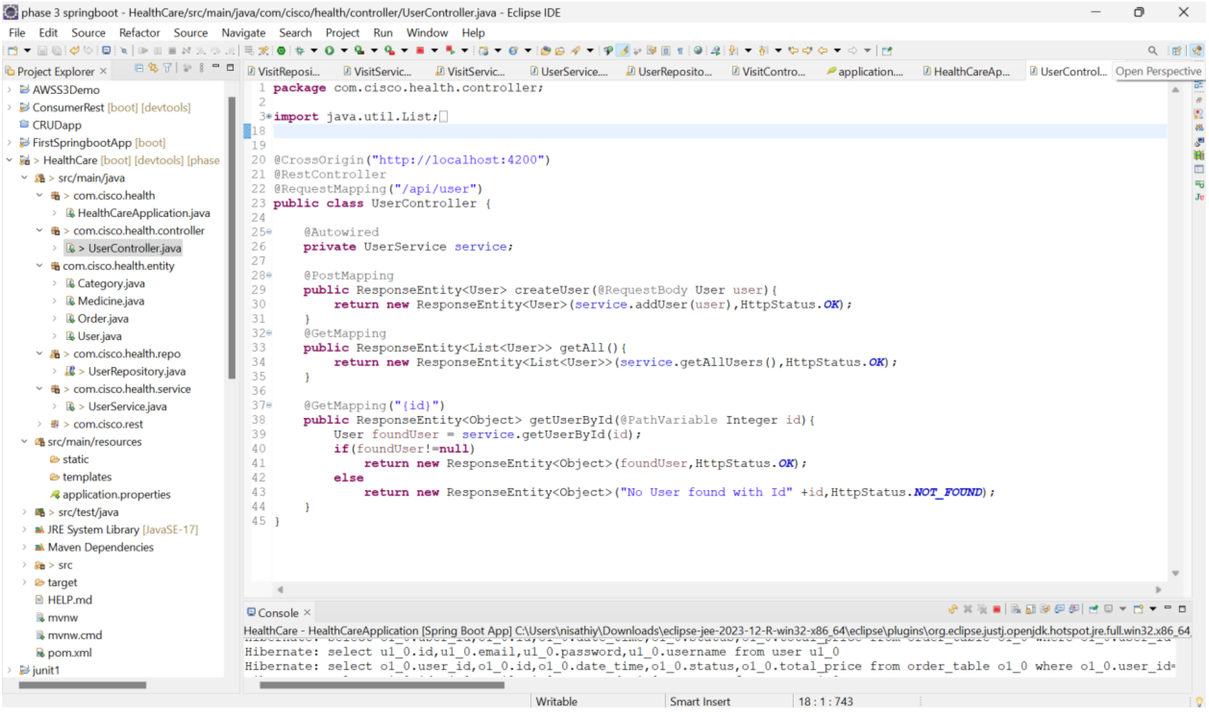


```typescript
import { TestBed } from '@angular/core/testing';
import { RouterTestingModule } from '@angular/router/testing';
import { AppComponent } from './app.component';

describe('AppComponent', () => {
  beforeEach(async () => {
    await TestBed.configureTestingModule({
      imports: [
        RouterTestingModule
      ],
      declarations: [
        AppComponent
      ],
    }).compileComponents();
  });

  it('should create the app', () => {
    const fixture = TestBed.createComponent(AppComponent);
    const app = fixture.componentInstance;
    expect(app).toBeTruthy();
  });

  it(`should have as title 'healthcare-frontend'`, () => {
    const fixture = TestBed.createComponent(AppComponent);
    const app = fixture.componentInstance;
    expect(app.title).toEqual('healthcare-frontend');
```

```
Build at: 2024-03-20T06:14:29.928Z - Hash: 3074fabc16962caf - Time: 545ms

√ Compiled successfully.
```

Backend Part in Eclipse:

User Controller:



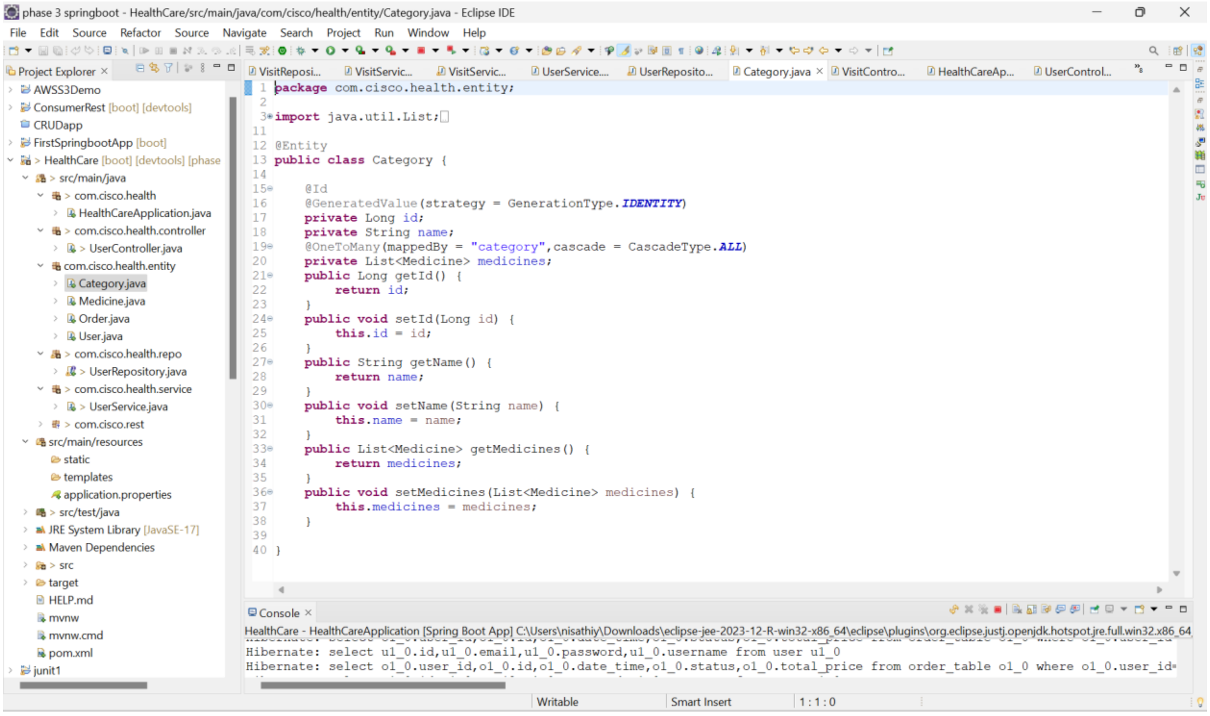
phase 3 springboot - HealthCare/src/main/java/com/cisco/health/controller/UserController.java - Eclipse IDE

```java
package com.cisco.health.controller;

import java.util.List;


@CrossOrigin("http://localhost:4200")
@RestController
@RequestMapping("/api/user")
public class UserController {

    @Autowired
    private UserService service;

    @PostMapping
    public ResponseEntity<User> createUser(@RequestBody User user){
        return new ResponseEntity<User>(service.addUser(user),HttpStatus.OK);
    }
    @GetMapping
    public ResponseEntity<List<User>> getAll(){
        return new ResponseEntity<List<User>>(service.getAllUsers(),HttpStatus.OK);
    }

    @GetMapping("{id}")
    public ResponseEntity<Object> getUserById(@PathVariable Integer id){
        User foundUser = service.getUserById(id);
        if(foundUser!=null)
            return new ResponseEntity<Object>(foundUser,HttpStatus.OK);
        else
            return new ResponseEntity<Object>("No User found with Id" +id,HttpStatus.NOT_FOUND);
    }
}
```

HealthCare - HealthCareApplication [Spring Boot App] C:\Users\nisathiy\Downloads\eclipse-jee-2023-12-R-win32-x86_64\eclipse\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64

```
Hibernate: select u1_0.id,u1_0.email,u1_0.password,u1_0.username from user u1_0
Hibernate: select o1_0.user_id,o1_0.id,o1_0.date_time,o1_0.status,o1_0.total_price from order_table o1_0 where o1_0.user_id=
```

Entity: Category, Medicine , Order and User.

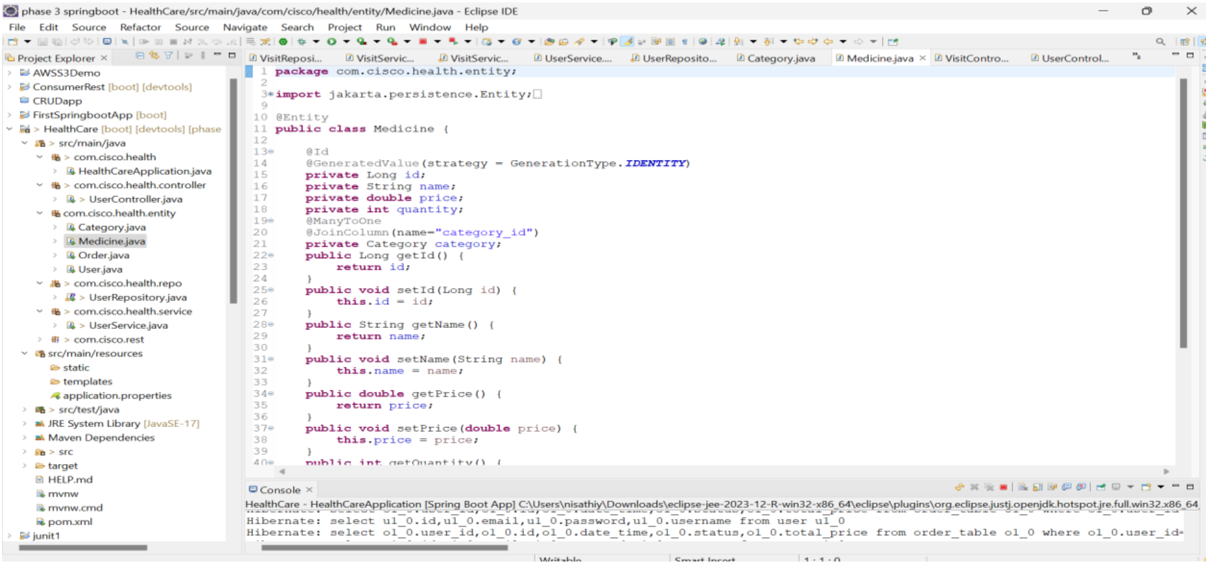


phase 3 springboot - HealthCare/src/main/java/com/cisco/health/entity/Category.java - Eclipse IDE

```java
package com.cisco.health.entity;

import java.util.List;

@Entity
public class Category {

    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private Long id;
    private String name;
    @OneToMany(mappedBy = "category",cascade = CascadeType.ALL)
    private List<Medicine> medicines;
    public Long getId() {
        return id;
    }
    public void setId(Long id) {
        this.id = id;
    }
    public String getName() {
        return name;
    }
    public void setName(String name) {
        this.name = name;
    }
    public List<Medicine> getMedicines() {
        return medicines;
    }
    public void setMedicines(List<Medicine> medicines) {
        this.medicines = medicines;
    }

}
```

HealthCare - HealthCareApplication [Spring Boot App] C:\Users\nisathiy\Downloads\eclipse-jee-2023-12-R-win32-x86_64\eclipse\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64

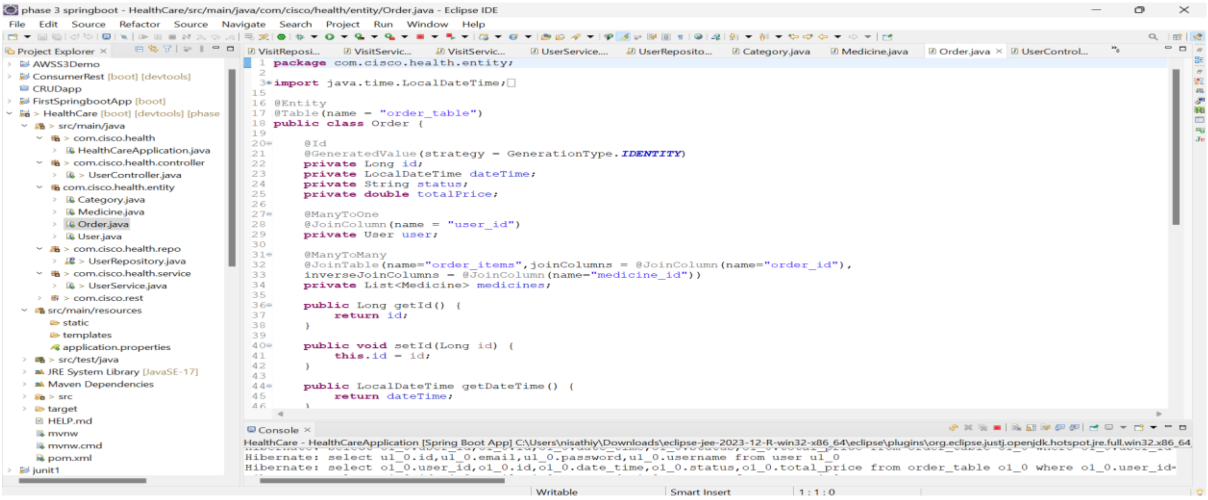
```
Hibernate: select u1_0.id,u1_0.email,u1_0.password,u1_0.username from user u1_0
Hibernate: select o1_0.user_id,o1_0.id,o1_0.date_time,o1_0.status,o1_0.total_price from order_table o1_0 where o1_0.user_id=
```



phase 3 springboot - HealthCare/src/main/java/com/cisco/health/entity/Medicine.java - Eclipse IDE

```java
package com.cisco.health.entity;

import jakarta.persistence.Entity;

@Entity
public class Medicine {

    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private Long id;
    private String name;
    private double price;
    private int quantity;
    @ManyToOne
    @JoinColumn(name="category_id")
    private Category category;
    public Long getId() {
        return id;
    }
    public void setId(Long id) {
        this.id = id;
    }
    public String getName() {
        return name;
    }
    public void setName(String name) {
        this.name = name;
    }
    public double getPrice() {
        return price;
    }
    public void setPrice(double price) {
        this.price = price;
    }
    public int getQuantity() {
```

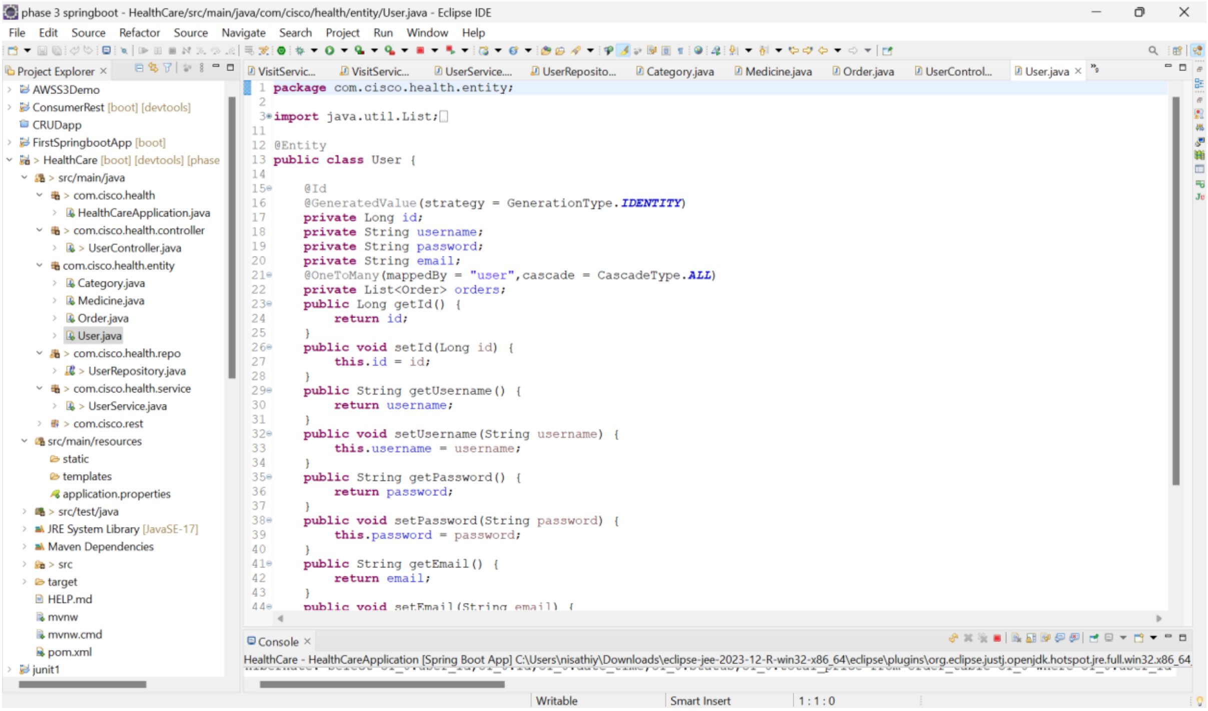
```
HealthCare - HealthCareApplication [Spring Boot App] C:\Users\nisathiy\Downloads\eclipse-jee-2023-12-R-win32-x86_64\eclipse\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64
Hibernate: select ul_0.id,ul_0.email,ul_0.password,ul_0.username from user ul_0
Hibernate: select ol_0.user_id,ol_0.id,ol_0.date_time,ol_0.status,ol_0.total_price from order_table ol_0 where ol_0.user_id=
```



phase 3 springboot - HealthCare/src/main/java/com/cisco/health/entity/Order.java - Eclipse IDE

```java
package com.cisco.health.entity;

import java.time.LocalDateTime;

@Entity
@Table(name = "order_table")
public class Order {

    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private Long id;
    private LocalDateTime dateTime;
    private String status;
    private double totalPrice;

    @ManyToOne
    @JoinColumn(name = "user_id")
    private User user;

    @ManyToMany
    @JoinTable(name="order_items",joinColumns = @JoinColumn(name="order_id"),
    inverseJoinColumns = @JoinColumn(name="medicine_id"))
    private List<Medicine> medicines;

    public Long getId() {
        return id;
    }

    public void setId(Long id) {
        this.id = id;
    }

    public LocalDateTime getDateTime() {
        return dateTime;
```

```
HealthCare - HealthCareApplication [Spring Boot App] C:\Users\nisathiy\Downloads\eclipse-jee-2023-12-R-win32-x86_64\eclipse\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64
Hibernate: select ul_0.id,ul_0.email,ul_0.password,ul_0.username from user ul_0
Hibernate: select ol_0.user_id,ol_0.id,ol_0.date_time,ol_0.status,ol_0.total_price from order_table ol_0 where ol_0.user_id=
```

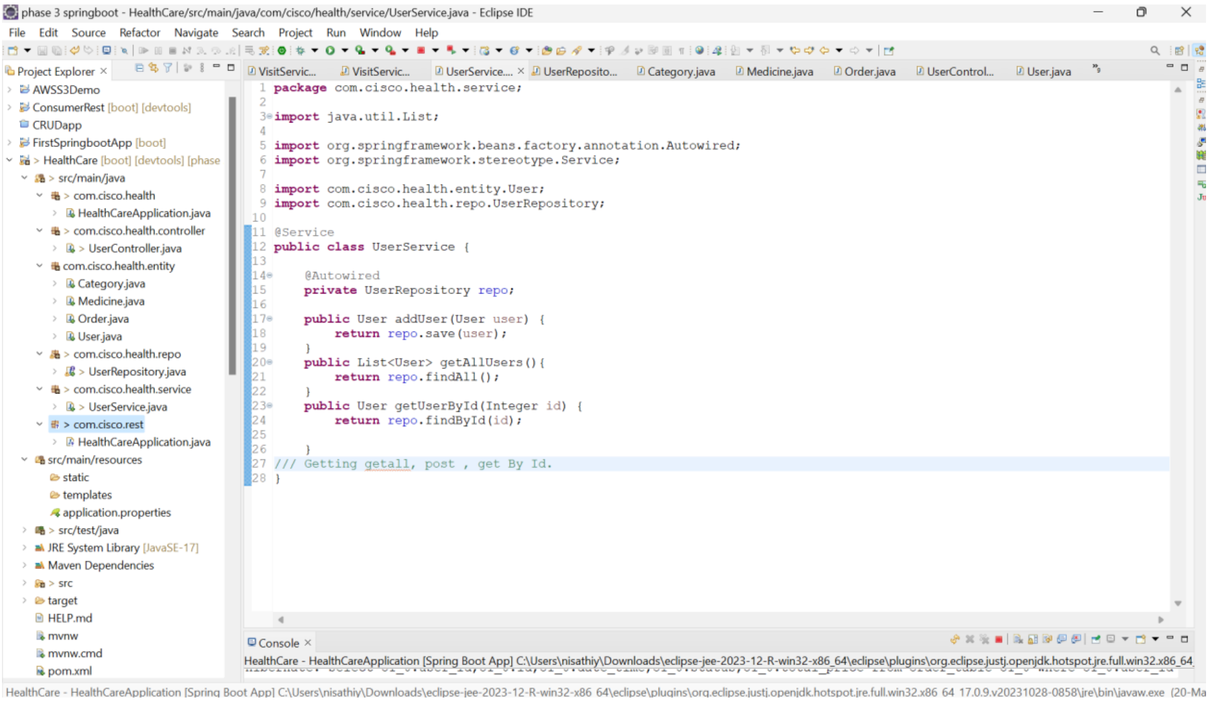


phase 3 springboot - HealthCare/src/main/java/com/cisco/health/entity/User.java - Eclipse IDE

```java
package com.cisco.health.entity;

import java.util.List;

@Entity
public class User {

    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private Long id;
    private String username;
    private String password;
    private String email;
    @OneToMany(mappedBy = "user",cascade = CascadeType.ALL)
    private List<Order> orders;
    public Long getId() {
        return id;
    }
    public void setId(Long id) {
        this.id = id;
    }
    public String getUsername() {
        return username;
    }
    public void setUsername(String username) {
        this.username = username;
    }
    public String getPassword() {
        return password;
    }
    public void setPassword(String password) {
        this.password = password;
    }
    public String getEmail() {
        return email;
    }
    public void setEmail(String email) {
```

```
HealthCare - HealthCareApplication [Spring Boot App] C:\Users\nisathiy\Downloads\eclipse-jee-2023-12-R-win32-x86_64\eclipse\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64
```

Service:

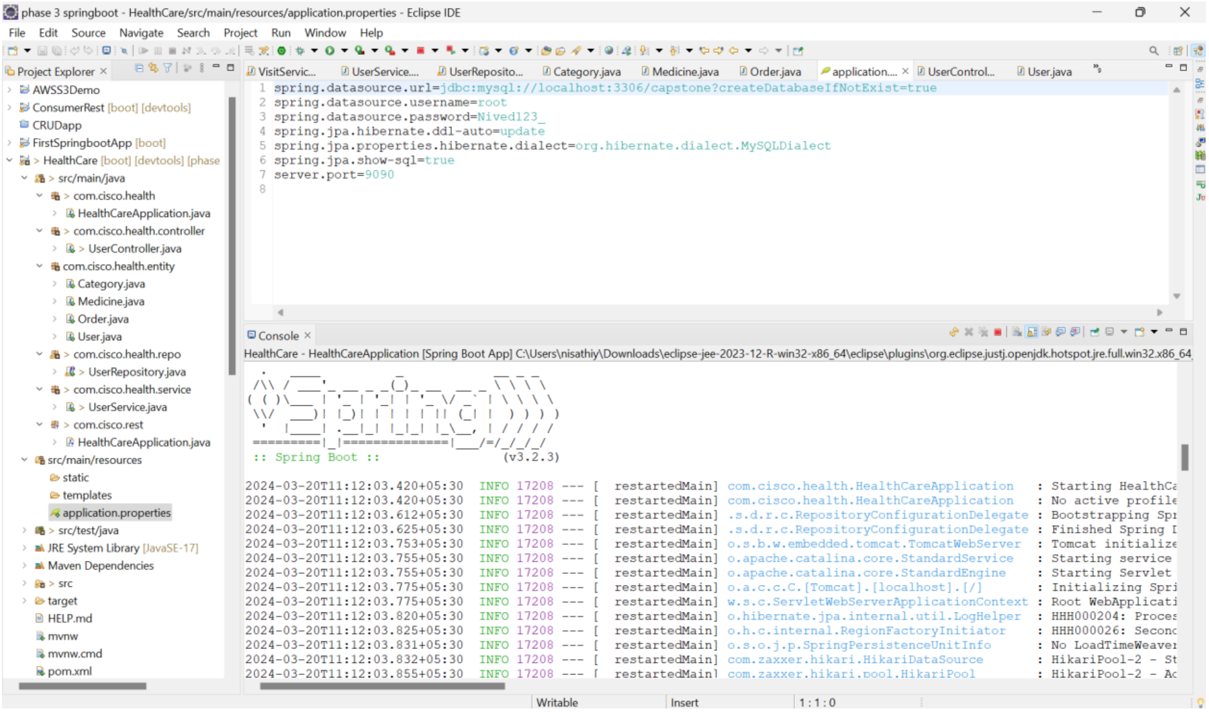


phase 3 springboot - HealthCare/src/main/java/com/cisco/health/service/UserService.java - Eclipse IDE

```java
package com.cisco.health.service;

import java.util.List;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Service;

import com.cisco.health.entity.User;
import com.cisco.health.repo.UserRepository;

@Service
public class UserService {

    @Autowired
    private UserRepository repo;

    public User addUser(User user) {
        return repo.save(user);
    }
    public List<User> getAllUsers(){
        return repo.findAll();
    }
    public User getUserById(Integer id) {
        return repo.findById(id);

    }
/// Getting getall, post , get By Id.
}
```

```
HealthCare - HealthCareApplication [Spring Boot App] C:\Users\nisathiy\Downloads\eclipse-jee-2023-12-R-win32-x86_64\eclipse\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64
HealthCare - HealthCareApplication [Spring Boot App] C:\Users\nisathiy\Downloads\eclipse-jee-2023-12-R-win32-x86_64\eclipse\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_17.0.9.v20231028-0858\jre\bin\javaw.exe (20-Ma
```

Application Properties:

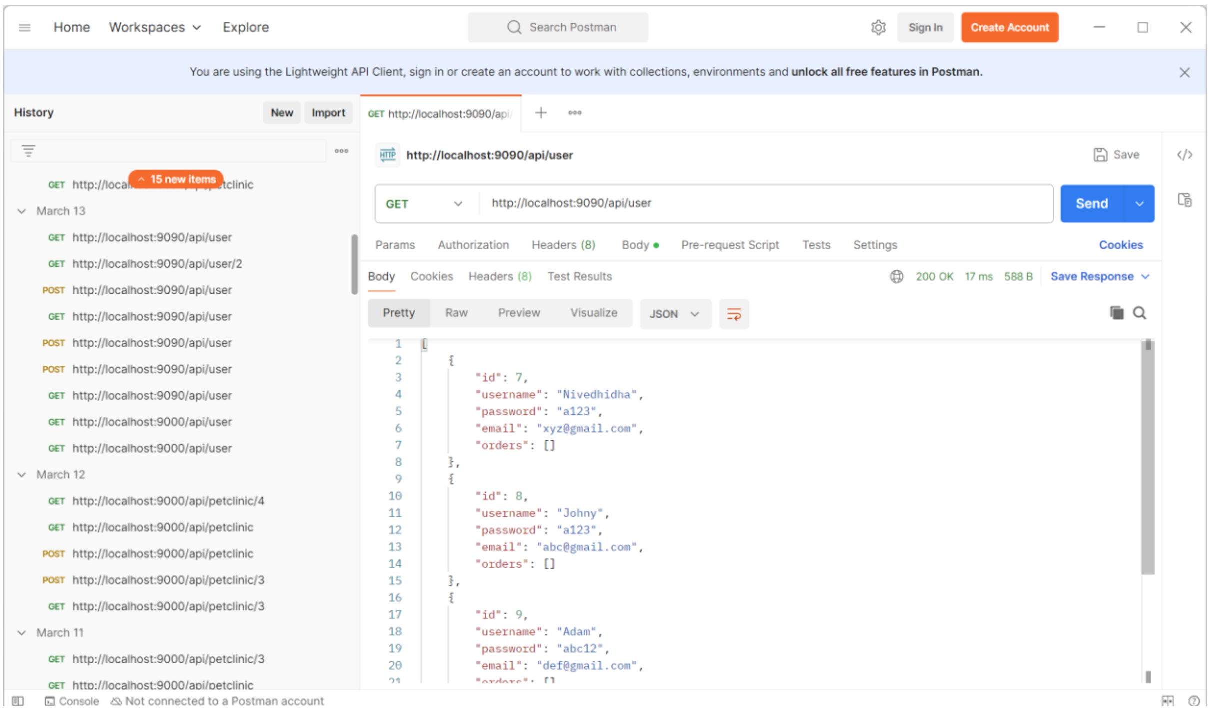


phase 3 springboot - HealthCare/src/main/resources/application.properties - Eclipse IDE

```
spring.datasource.url=jdbc:mysql://localhost:3306/capstone?createDatabaseIfNotExist=true
spring.datasource.username=root
spring.datasource.password=Nived123_
spring.jpa.hibernate.ddl-auto=update
spring.jpa.properties.hibernate.dialect=org.hibernate.dialect.MySQLDialect
spring.jpa.show-sql=true
server.port=9090
```

```
HealthCare - HealthCareApplication [Spring Boot App] C:\Users\nisathiy\Downloads\eclipse-jee-2023-12-R-win32-x86_64\eclipse\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64

  .   ____          _            __ _ _
 /\\ / ___'_ __ _ _(_)_ __  __ _ \ \ \ \
( ( )\___ | '_ | '_| | '_ \/ _` | \ \ \ \
 \\/  ___)| |_)| | | | | || (_| |  ) ) ) )
  '  |____| .__|_| |_|_| |_\__, | / / / /
 =========|_|==============|___/=/_/_/_/
 :: Spring Boot ::                (v3.2.3)

2024-03-20T11:12:03.420+05:30  INFO 17208 --- [  restartedMain] com.cisco.health.HealthCareApplication   : Starting HealthCa
2024-03-20T11:12:03.420+05:30  INFO 17208 --- [  restartedMain] com.cisco.health.HealthCareApplication   : No active profile
2024-03-20T11:12:03.612+05:30  INFO 17208 --- [  restartedMain] .s.d.r.c.RepositoryConfigurationDelegate : Bootstrapping Spr
2024-03-20T11:12:03.625+05:30  INFO 17208 --- [  restartedMain] .s.d.r.c.RepositoryConfigurationDelegate : Finished Spring I
2024-03-20T11:12:03.753+05:30  INFO 17208 --- [  restartedMain] o.s.b.w.embedded.tomcat.TomcatWebServer  : Tomcat initialize
2024-03-20T11:12:03.755+05:30  INFO 17208 --- [  restartedMain] o.apache.catalina.core.StandardService   : Starting service
2024-03-20T11:12:03.755+05:30  INFO 17208 --- [  restartedMain] o.apache.catalina.core.StandardEngine    : Starting Servlet
2024-03-20T11:12:03.775+05:30  INFO 17208 --- [  restartedMain] o.a.c.c.C.[Tomcat].[localhost].[/]       : Initializing Spri
2024-03-20T11:12:03.775+05:30  INFO 17208 --- [  restartedMain] w.s.c.ServletWebServerApplicationContext : Root WebApplicati
2024-03-20T11:12:03.820+05:30  INFO 17208 --- [  restartedMain] o.hibernate.jpa.internal.util.LogHelper  : HHH000204: Proces
2024-03-20T11:12:03.825+05:30  INFO 17208 --- [  restartedMain] o.h.c.internal.RegionFactoryInitiator    : HHH000026: Seconc
2024-03-20T11:12:03.831+05:30  INFO 17208 --- [  restartedMain] o.s.o.j.p.SpringPersistenceUnitInfo      : No LoadTimeWeaver
2024-03-20T11:12:03.832+05:30  INFO 17208 --- [  restartedMain] com.zaxxer.hikari.HikariDataSource       : HikariPool-2 - St
2024-03-20T11:12:03.855+05:30  INFO 17208 --- [  restartedMain] com.zaxxer.hikari.pool.HikariPool        : HikariPool-2 - Ac
```

Get All API Response:



You are using the Lightweight API Client, sign in or create an account to work with collections, environments and **unlock all free features in Postman.**

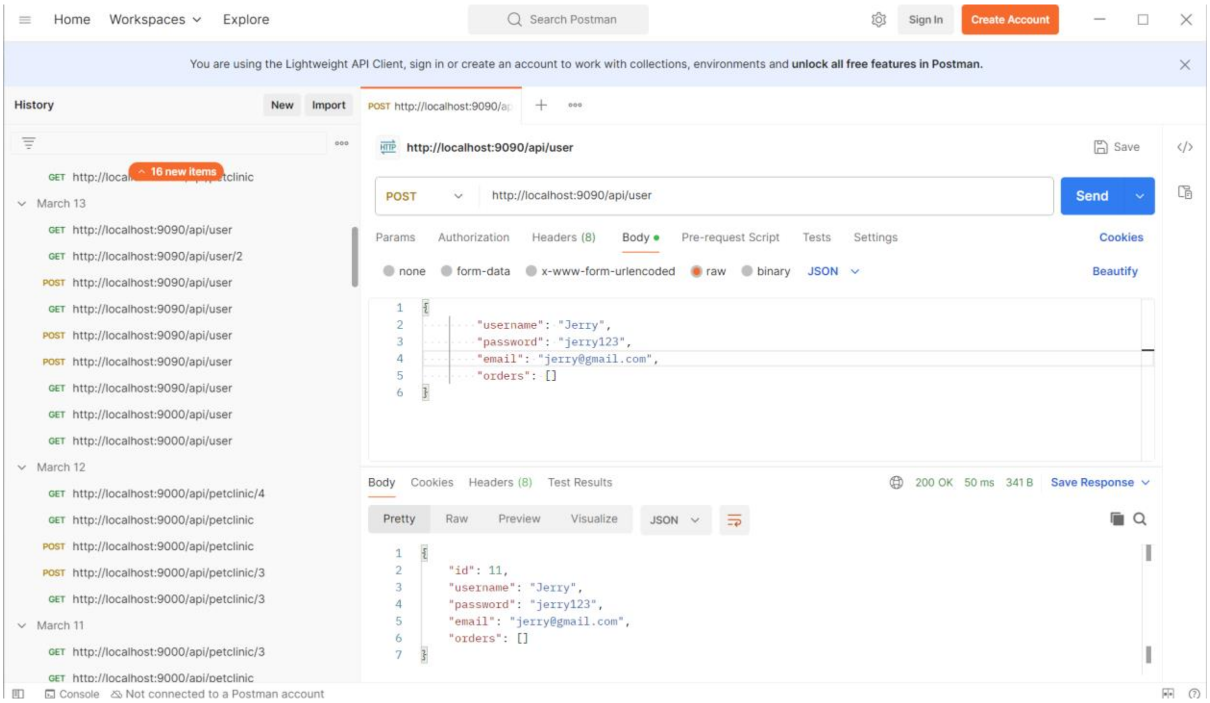
GET http://localhost:9090/api/user

200 OK 17 ms 588 B

```
GET http://local...etclinic
March 13
GET http://localhost:9090/api/user
GET http://localhost:9090/api/user/2
POST http://localhost:9090/api/user
GET http://localhost:9090/api/user
POST http://localhost:9090/api/user
POST http://localhost:9090/api/user
GET http://localhost:9090/api/user
GET http://localhost:9000/api/user
GET http://localhost:9000/api/user
March 12
GET http://localhost:9000/api/petclinic/4
GET http://localhost:9000/api/petclinic
POST http://localhost:9000/api/petclinic
POST http://localhost:9000/api/petclinic/3
GET http://localhost:9000/api/petclinic/3
March 11
GET http://localhost:9000/api/petclinic/3
GET http://localhost:9000/api/petclinic
```

```
[
    {
        "id": 7,
        "username": "Nivedhidha",
        "password": "a123",
        "email": "xyz@gmail.com",
        "orders": []
    },
    {
        "id": 8,
        "username": "Johny",
        "password": "a123",
        "email": "abc@gmail.com",
        "orders": []
    },
    {
        "id": 9,
        "username": "Adam",
        "password": "abc12",
        "email": "def@gmail.com",
        "orders": []
```

Post Object:



You are using the Lightweight API Client, sign in or create an account to work with collections, environments and **unlock all free features in Postman.**

POST http://localhost:9090/api/user

○ none ○ form-data ○ x-www-form-urlencoded ● raw ○ binary JSON

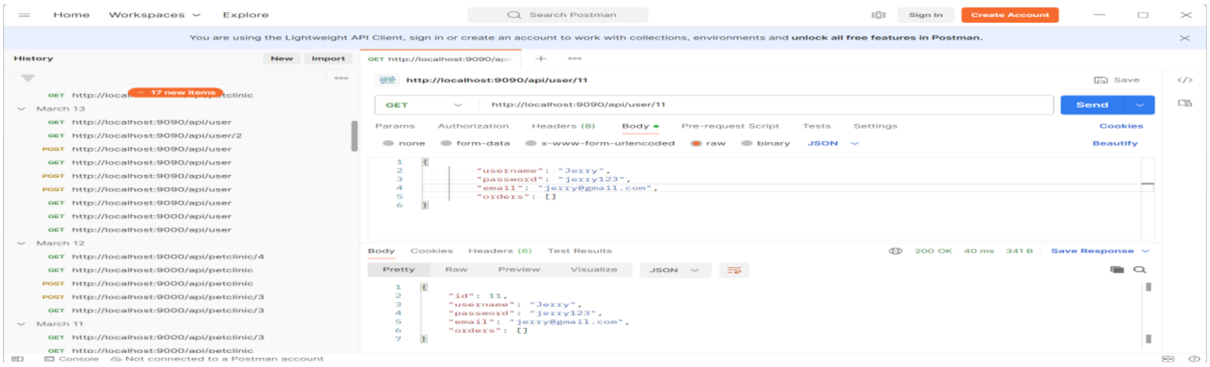
```
{
    "username": "Jerry",
    "password": "jerry123",
    "email": "jerry@gmail.com",
    "orders": []
}
```

200 OK 50 ms 341 B

```
{
    "id": 11,
    "username": "Jerry",
    "password": "jerry123",
    "email": "jerry@gmail.com",
    "orders": []
}
```

```
GET http://local...etclinic
March 13
GET http://localhost:9090/api/user
GET http://localhost:9090/api/user/2
POST http://localhost:9090/api/user
GET http://localhost:9090/api/user
POST http://localhost:9090/api/user
POST http://localhost:9090/api/user
GET http://localhost:9090/api/user
GET http://localhost:9000/api/user
GET http://localhost:9000/api/user
March 12
GET http://localhost:9000/api/petclinic/4
GET http://localhost:9000/api/petclinic
POST http://localhost:9000/api/petclinic
POST http://localhost:9000/api/petclinic/3
GET http://localhost:9000/api/petclinic/3
March 11
GET http://localhost:9000/api/petclinic/3
GET http://localhost:9000/api/petclinic
```

Get Object by Id:



You are using the Lightweight API Client, sign in or create an account to work with collections, environments and **unlock all free features in Postman.**

History

GET http://local... ...etclinic

March 13

- GET http://localhost:9090/api/user
- GET http://localhost:9090/api/user/2
- POST http://localhost:9090/api/user
- GET http://localhost:9090/api/user
- POST http://localhost:9090/api/user
- POST http://localhost:9090/api/user
- GET http://localhost:9090/api/user
- GET http://localhost:9000/api/user
- GET http://localhost:9000/api/user

March 12

- GET http://localhost:9000/api/petclinic/4
- GET http://localhost:9000/api/petclinic
- POST http://localhost:9000/api/petclinic
- POST http://localhost:9000/api/petclinic/3
- GET http://localhost:9000/api/petclinic/3

March 11

- GET http://localhost:9000/api/petclinic/3
- GET http://localhost:9000/api/petclinic

http://localhost:9090/api/user/11

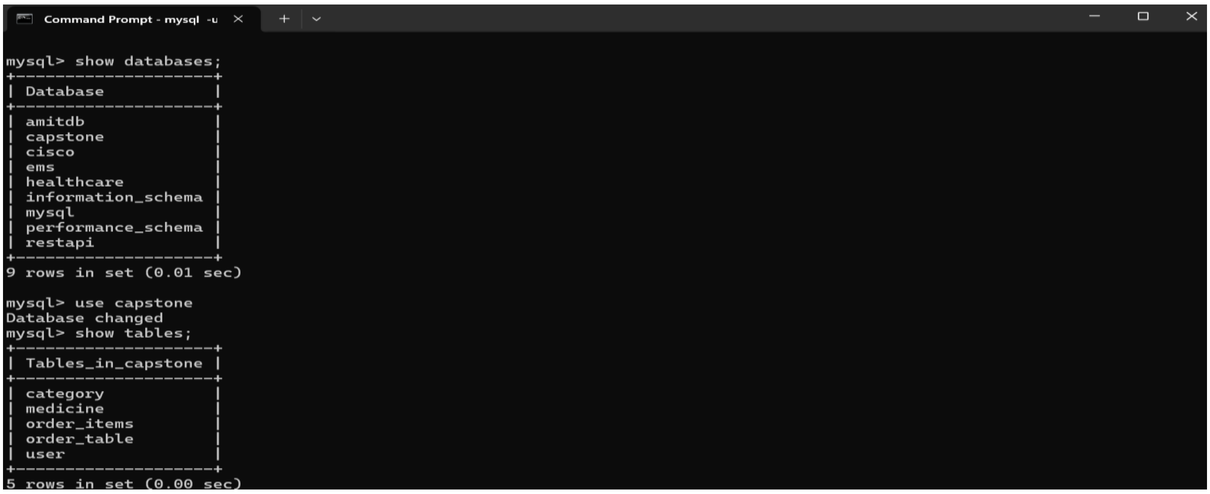
GET http://localhost:9090/api/user/11

```
{
    "username": "Jerry",
    "password": "jerry123",
    "email": "jerry@gmail.com",
    "orders": []
}
```

200 OK 40 ms 341 B

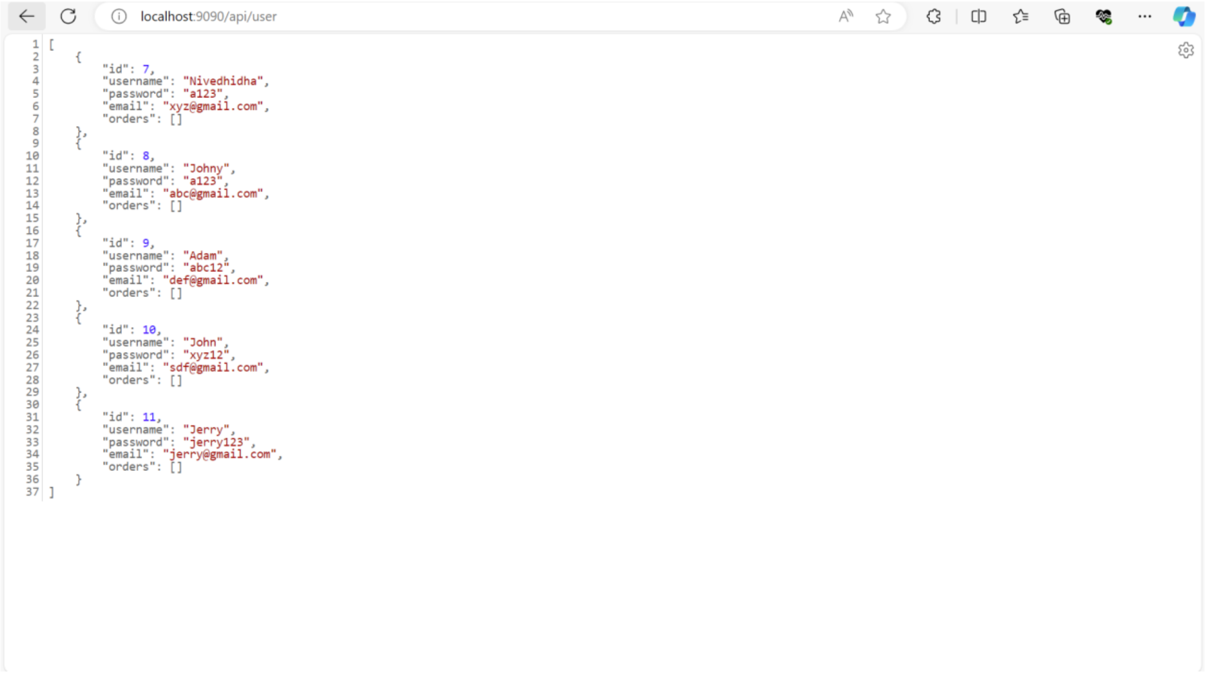
```
{
    "id": 11,
    "username": "Jerry",
    "password": "jerry123",
    "email": "jerry@gmail.com",
    "orders": []
}
```

Database: Capstone.



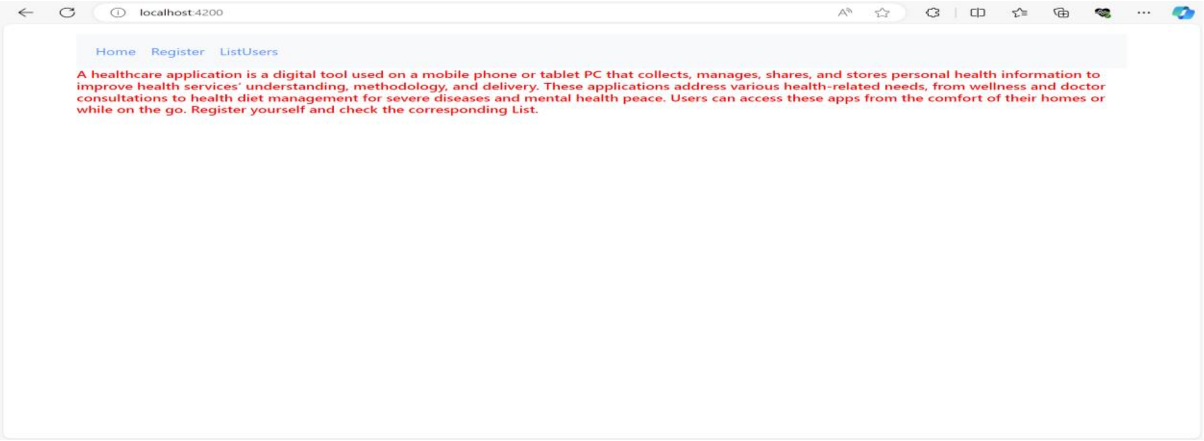
```
mysql> show databases;
+--------------------+
| Database           |
+--------------------+
| amitdb             |
| capstone           |
| cisco              |
| ems                |
| healthcare         |
| information_schema |
| mysql              |
| performance_schema |
| restapi            |
+--------------------+
9 rows in set (0.01 sec)

mysql> use capstone
Database changed
mysql> show tables;
+--------------------+
| Tables_in_capstone |
+--------------------+
| category           |
| medicine           |
| order_items        |
| order_table        |
| user               |
+--------------------+
5 rows in set (0.00 sec)
```

Local Host Response:



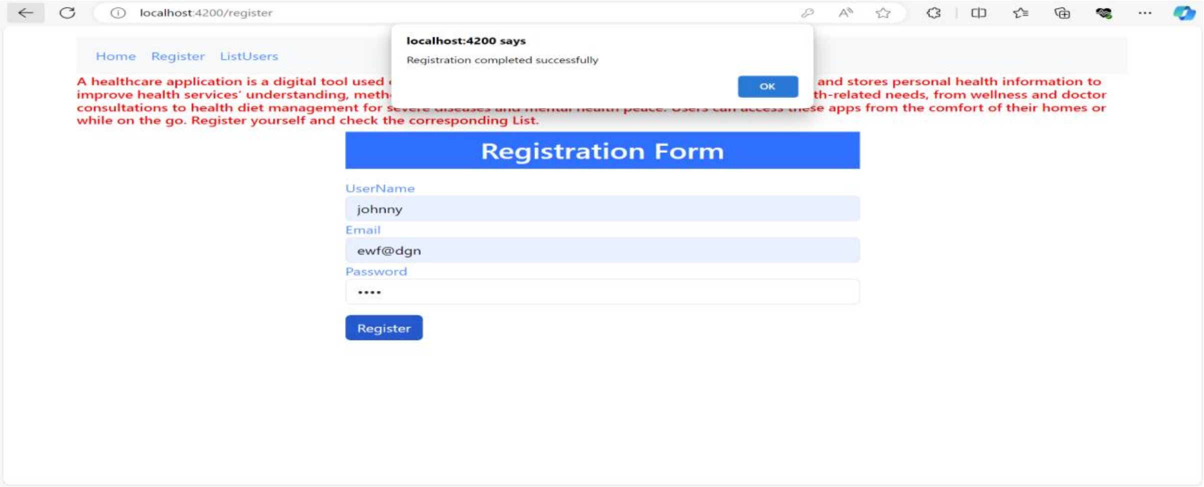
```
[
    {
        "id": 7,
        "username": "Nivedhidha",
        "password": "a123",
        "email": "xyz@gmail.com",
        "orders": []
    },
    {
        "id": 8,
        "username": "Johny",
        "password": "a123",
        "email": "abc@gmail.com",
        "orders": []
    },
    {
        "id": 9,
        "username": "Adam",
        "password": "abc12",
        "email": "def@gmail.com",
        "orders": []
    },
    {
        "id": 10,
        "username": "John",
        "password": "xyz12",
        "email": "sdf@gmail.com",
        "orders": []
    },
    {
        "id": 11,
        "username": "Jerry",
        "password": "jerry123",
        "email": "jerry@gmail.com",
        "orders": []
    }
]
```

Output Screen.



A healthcare application is a digital tool used on a mobile phone or tablet PC that collects, manages, shares, and stores personal health information to improve health services' understanding, methodology, and delivery. These applications address various health-related needs, from wellness and doctor consultations to health diet management for severe diseases and mental health peace. Users can access these apps from the comfort of their homes or while on the go. Register yourself and check the corresponding List.

Registration Screen.



localhost:4200 says

Registration completed successfully

A healthcare application is a digital tool used ... and stores personal health information to improve health services' understanding, meth... th-related needs, from wellness and doctor consultations to health diet management for ... se apps from the comfort of their homes or while on the go. Register yourself and check the corresponding List.

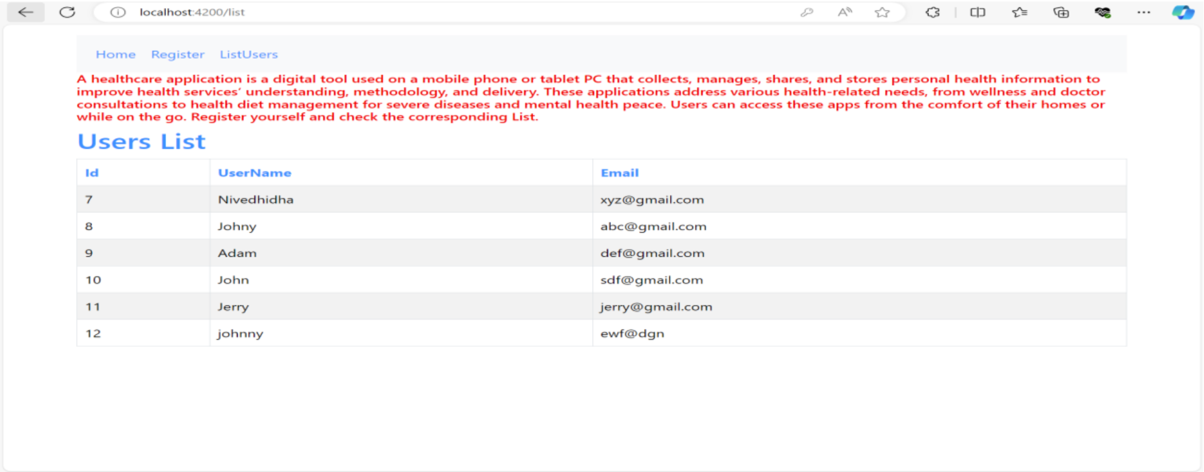
Registration Form

UserName: johnny

Email: ewf@dgn

Password: ••••

UserList.

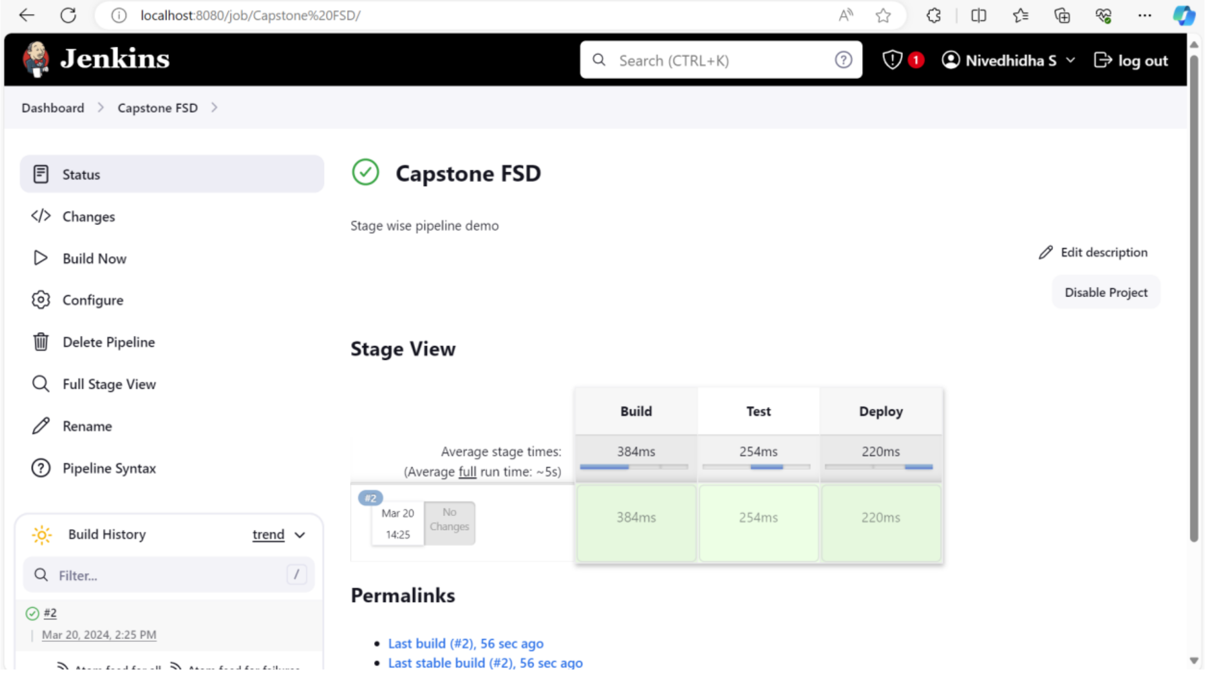


A healthcare application is a digital tool used on a mobile phone or tablet PC that collects, manages, shares, and stores personal health information to improve health services' understanding, methodology, and delivery. These applications address various health-related needs, from wellness and doctor consultations to health diet management for severe diseases and mental health peace. Users can access these apps from the comfort of their homes or while on the go. Register yourself and check the corresponding List.

Users List

Id	UserName	Email
7	Nivedhidha	xyz@gmail.com
8	Johny	abc@gmail.com
9	Adam	def@gmail.com
10	John	sdf@gmail.com
11	Jerry	jerry@gmail.com
12	johnny	ewf@dgn

Jenkins Capstone Project:



Capstone FSD

Stage wise pipeline demo

Stage View

	Build	Test	Deploy
Average stage times: (Average full run time: ~5s)	384ms	254ms	220ms
#2 Mar 20 14:25 No Changes	384ms	254ms	220ms

Permalinks

- Last build (#2), 56 sec ago
- Last stable build (#2), 56 sec ago