

Mini Project report on

Plate(Recipe search engine)

By

Meghna Wankhede

2020510068

Under the guidance of

Internal Supervisor

Dr. AARTI M. KARANDE



Department of Master of Computer Applications Sardar Patel

Institute of Technology Autonomous Institute Affiliated to

Mumbai University

2021-22

CERTIFICATE OF APPROVAL

This is to certify that the following students

Meghna Wankhede

2020510068

Have satisfactorily carried out work on the project entitled

“Plate (Recipe search engine)”

Towards the fulfillment of summer project, as laid down by University of Mumbai during
year 2021-22.

Project Guide
(Dr. Aarti Karande)

PROJECT APPROVAL CERTIFICATE

This is to certify that the following students

Meghna Wankhede

[2020510068]

Have successfully completed the Project report on “**Plate(Recipe search engine)** ”, which
is found to be satisfactory and is approved

At

SARDAR PATEL INSTITUTE OF TECHNOLOGY,
ANDHERI (W), MUMBAI.

INTERNAL EXAMINER
EXAMINER

EXTERNAL

Head of Department
(Dr. Pooja Raundale)

Principal
(Dr. B.N Chaudhari)

INDEX

Serial no.	Topic	Page no.
1.	Introduction	6
1.1.	Problem Definition	6
1.2.	Objective and Scope	6
1.3.	System Requirements	6
2.	Literature Survey	7
3.	SRS and Design	8
3.1.	Introduction	8
3.1.1.	Purpose	8
3.1.2.	System overview	8
3.2.	Overall Description	8
3.3.	Specific Requirements	8
3.3.1.	External interface requirements	9
3.3.2.	Functional requirements	9
4.	Project Analysis and Design	10
4.1.	Methodologies Accepted	10
4.1.1.	Detailed life cycle of the project	11
4.2.	UML Diagram	12
4.2.1.	Use case diagram	12
4.2.2.	Sequence diagram	14
4.2.3.	Flowchart diagram	15
5.	Project Implementation and Testing	16
5.1.	Work Break Down Structure	16
5.2.	Gantt Chart	16
5.3.	Pert Chart	17

5.4.	Code with reference to design	17
5.5.	Snapshot of UI	49
5.6.	Test Cases and Report	55
5.6.1.	Test cases report and results	55
6.	Documentation & Installation	56
7.	Future Enhancements	57
8.	Limitations	58
9.	Conclusion	59
10.	Bibliography.....	60

Chapter 1

INTRODUCTION

1.1 Problem Statement

To design an application through which users can select the leftover ingredients and find possible recipes of dishes.

1.2 Objective and Scope

1.2.1 Objectives:

1. To develop an application through users can select ingredients to find recipes and save them.
2. To develop an application with login – Users.

1.2.2 Scope:

1. To achieve the Primary goal of this application i.e. the user selects or adds ingredients to find possible recipes.
2. This application can be used by any one with app downloaded.
3. Firstly, all the new users have to register themselves and login in order to use the application.

1.3 System Requirements

1. Hardware Requirements:

- Standard I/O Devices
- Minimum 1GB RAM

2. Software Requirements:

- Browser
- Visual Studio Code Editor

Chapter 2

Literature Survey

Recipe retrieval is a representative and useful application of cross-modal information retrieval. Recent studies have proposed frameworks for retrieving images of cuisines given textual ingredient lists and instructions. However, the textual form of ingredients easily causes information loss or inaccurate description, especially for novices of cookery who are often the main users of recipe retrieval systems. [1]the task of recipe retrieval by taking images of ingredients as input queries, and retrieving cuisine images by incorporating visual information of ingredients through a deep convolutional neural network. [2]the utilization of pre-trained embeddings to perform recipe search and compare our search results with a keyword based search. The comparison the health score, nutritional content and recipe titles returned using both search approaches.[3] This paper proposes a new recipe search application using a knowledge-based spontaneous dialogue system to assistant users' operation on mobile devices.The proposed application asks the user a series of questions related to various cooking categories including recipe genres and cooking needs in order to narrow down the potential recipes that meet users' wants. A decision algorithm is proposed to find the best questions to be asked to narrow down the candidates as quickly as possible according to the recipe database.[4]In this paper a japanese group participated in the subtask involving an ad hoc Japanese recipe search. Their goal was to evaluate the effectiveness of our Japanese cooking ontology for the recipe search. To investigate the effectiveness of their ontology-based approach, they conducted experiments and found that their method can improve upon traditional document retrieval systems.[5] The aim of this paper approach is to support the detection of food categories in order to detect which one might be dangerous for a user affected by chronic disease. Their approach relies on background knowledge where recipes, food categories, and their relatedness with chronic diseases are modeled within a state-of-the-art ontology. Experiments conducted on a new publicly released dataset demonstrated the effectiveness of the proposed approach with respect to state-of-the-art classification strategies.

Chapter3

Software Requirement Specification (SRS) and Design

3.1 Introduction

- Food Recipe recommendation system based on available ingredients.
- Category wise ingredient listing for selecting the ingredients for recipe search as per available items.
- Cuisine wise recipe listing.
- Saved Recipes.
- Recipe search by name of the Dish.

3.1.1 Purpose

- The purpose of this SRS Document is to present a description of project. This SRS outlines the process followed to gather the requirements for the project. This document will also describe how the requirement statements gathered from the stakeholders make their way into features of the system.
- This document will, in addition, explain the scope, interfaces, and features as well as graphically describe the processes, functions and phases of the Software Development Life-cycle.

3.2 Overall Description

In this application user have to login in the application to access the features of the application. After login, User can view the favourites saved in their account. They can add ingredients and search for recipes and save them as their favourites in their account..

3.3 Specific Requirements

3.3.1 Functional Requirements

- The website should validate each and every user.
- At the time of login, the database must check if a user already have an account.
- The user can only login if already registered or he/she is a registered member.

3.3.2 Non-functional Requirements

- User should know what category their ingredients belong by them.
- Reliable

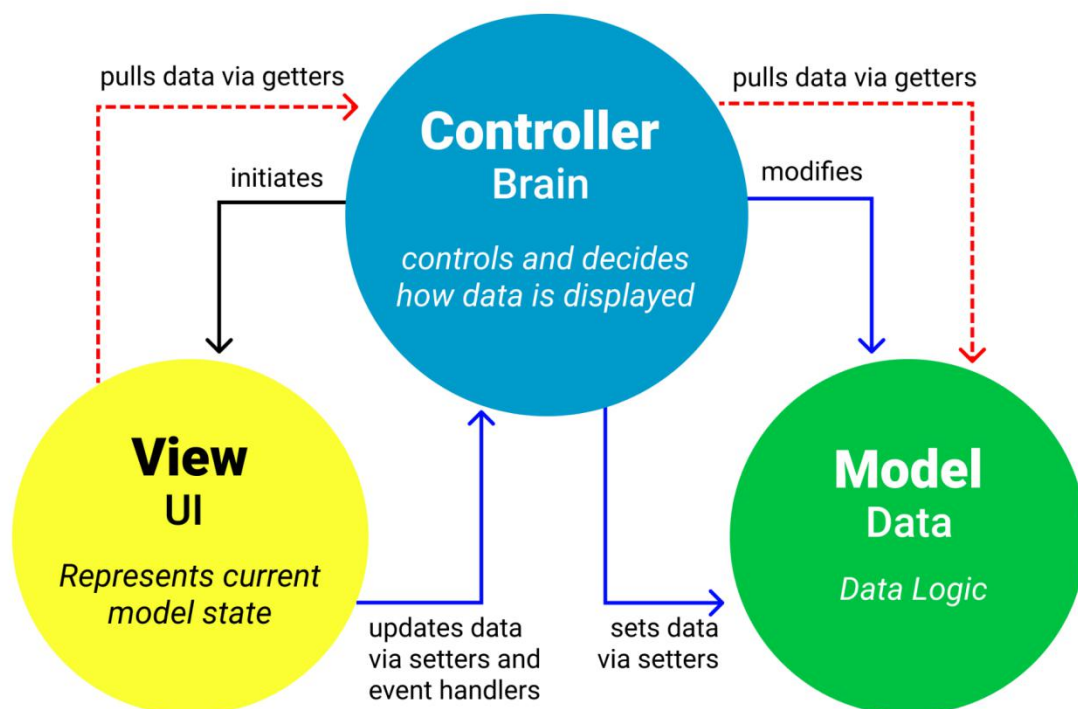
Chapter 4

Project Analysis and Design

4.1 Methodologies Adapted

The MVC pattern helps you break up the frontend and backend code into separate components. This way, it's much easier to manage and make changes to either side without them interfering with each other.

MVC Architecture Pattern



4.1.1 Detailed life cycle of the project.

- **Requirement Gathering and analysis** – All possible requirements of the system to be developed are captured in this phase and documented in a requirement specification document.
- **System Design** – The requirement specifications from first phase are studied in this phase and the system design is prepared. This system design helps in specifying hardware and system requirements and helps in defining the overall system architecture.
- **Implementation** – With inputs from the system design, the system is first developed in small programs called units, which are integrated in the next phase. Each unit is developed and tested for its functionality, which is referred to as Unit Testing.
- **Integration and Testing** – All the units developed in the implementation phase are integrated into a system after testing of each unit. Post integration the entire system is tested for any faults and failures.
- **Deployment of system** – Once the functional and non-functional testing is done; the product is deployed in the customer environment or released into the market.
- **Maintenance** – There are some issues which come up in the client environment. To fix those issues, patches are released. Also to enhance the product some better versions are released. Maintenance is done to deliver these changes in the customer environment.

4.2 UML Diagram

4.2.1 Use Case Diagram

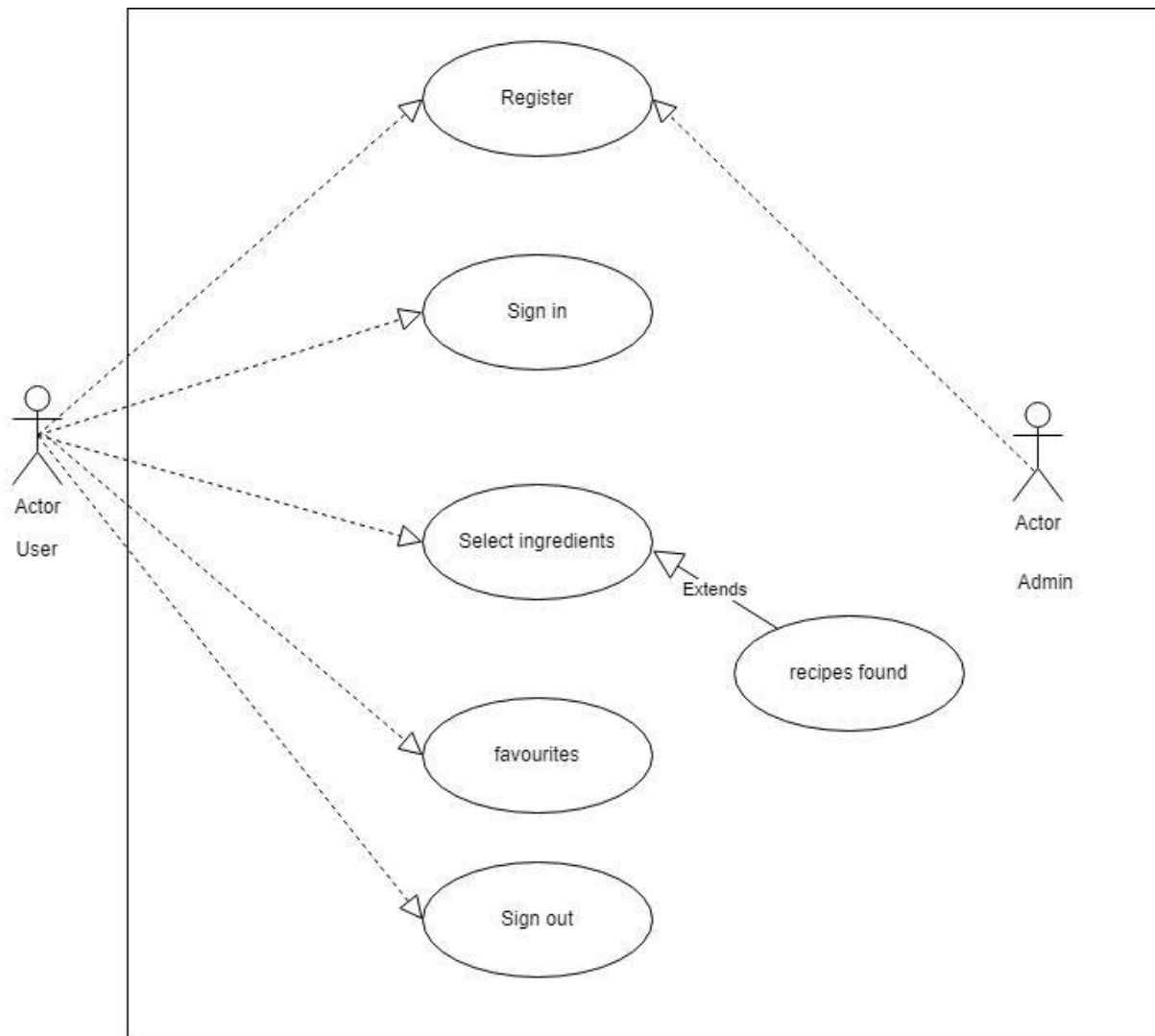


Fig no. 4.2.1 Use case diagram of Recipe search engine (plate). It shows all the features it will cover.

Specification table of use case diagram

Actor:	<ol style="list-style-type: none"> 1. User 2. Admin
Description of use cases:	<ol style="list-style-type: none"> 1. Register – User will register through this. Admin will also register through this. 2. Login – User and admin will login through this. 3. Select Ingredients – User can select the ingredients through this. 4. Recipe found – User can find the possible recipes here. 5. Favourites – User can view their favourite recipes saved In favorites section 6. Logout – Through this User and Admin can logout from the application.
Preconditions:	The user has to be authenticated.
Extends:	Recipe found

4.2.2 Sequence Diagram

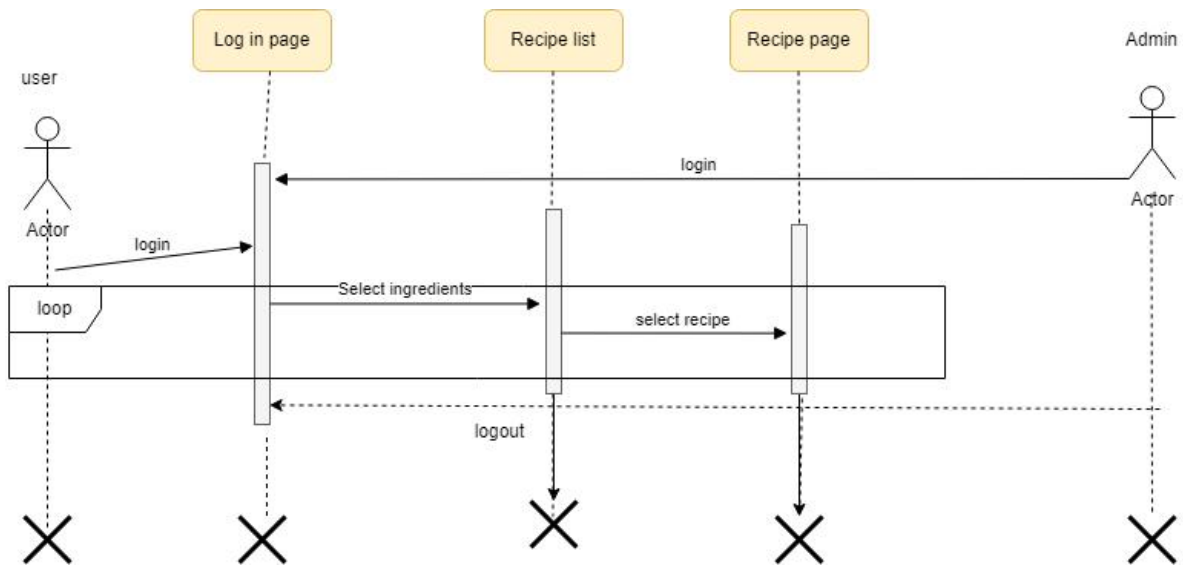


Fig no.: 4.2.2 Sequence diagram of Plate to show the process of the application

S.No.	Articrafts	Description
1.	Objects	LoginPage, RecipelistPage, Recipelist
2.	Asynchronous messages	<ol style="list-style-type: none"> 1. Login to the application 2. Select ingredients after entering the HomePage
3.	Synchronous messages	<ol style="list-style-type: none"> 1. Logout for User 2. Logout for Corporator
4.	Types of frames	Loop
5.	Actors	User, Admin

4.2.3 Flowchart Diagram

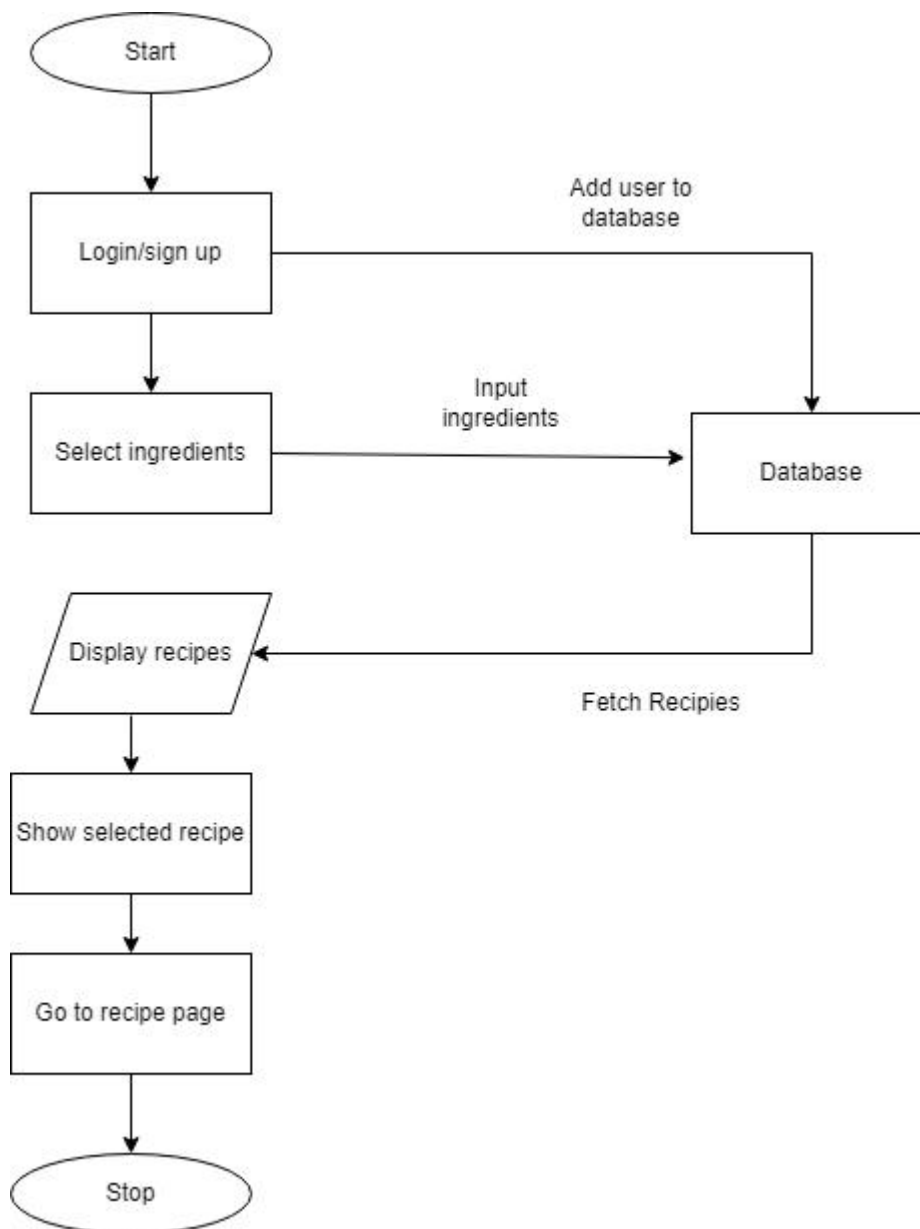


Fig no. 4.2.3 Flowchart diagram of Plate.

Chapter 5

Project Implementation and Testing

5.1 Work Breakdown Structure

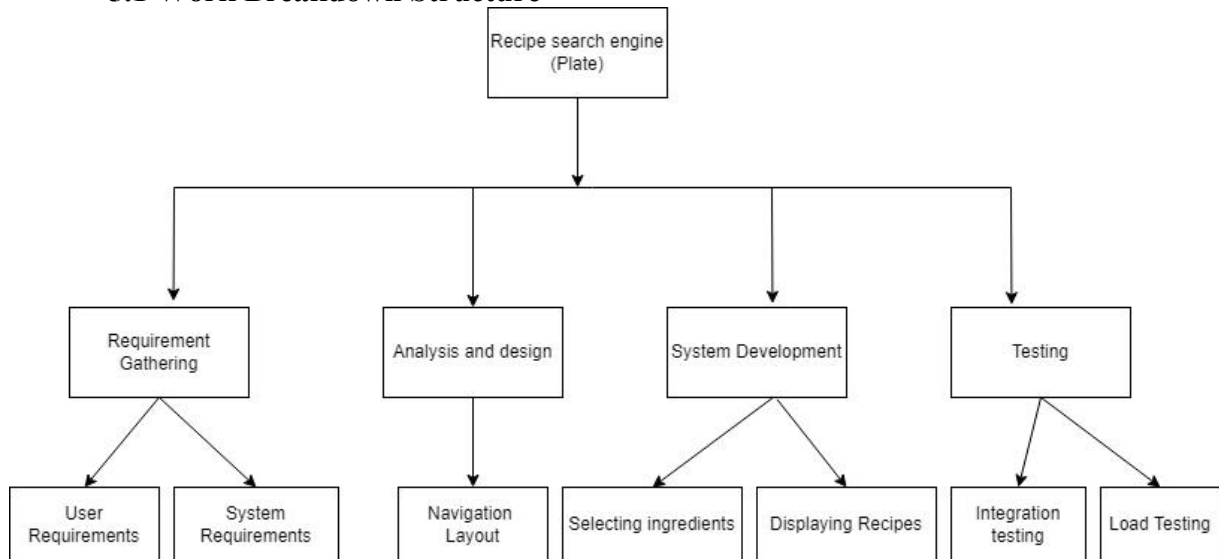


Fig no. 5.1 Work Breakdown Structure for the Plate.

5.2 Gantt Chart



Fig no. 5.2 Gantt Chart for the Plate.

5.4 Code with reference to design

NodeJS code:

index.js

```
var modules=require("./modules");
modules.services=require('./services')(modules);
module.routes=require('./routes')(modules);
modules.server.listen(modules.config.server_port,function(){
    console.log("Server Started");
});
```

config.js

```
var config={
    mongodbConfig:{
        url:'mongodb://127.0.0.1:27017',
        dbname:'recipe',
        useUrlParser: true ,
        server_url:'127.0.0.1',
        port:27017
    },
    server_port: 3000,
};

module.exports=config
```

modules.js

```
var modules={
    express: require('express'),
    http: require('http'),
    moment: require('moment'),
    bodyParser: require('body-parser'),
    request: require('request'),
    fs: require('fs'),
    cors: require('cors'),
    mongodb: require('mongodb'),
    mongo: require('mongodb').MongoClient,
    objectid: require('mongodb').ObjectID,
    rp: require('request-promise'),
    recipeScrape: require("recipe-scraper"),
};

modules.config=require('./config');

modules.mongo.connect(modules.config.mongodbConfig.url,{ useUrlParser
true },function(err,client){
    if(!err){
```

```
modules.db=client.db(modules.config.mongodbConfig.dbname);  
// modules.grid_server=modules.grid(modules.db,modules.mongodb);  
}  
else{  
    console.log(err)  
}  
})  
  
modules.app=modules.express();  
  
modules.app.use(modules.bodyParser.json({limit: '50mb'}));  
  
// modules.app=modules.express();  
  
modules.app.use(modules.cors());  
  
modules.app.use(function(req,res,next){  
    res.header("Access-Control-Allow-Origin", "*");  
    res.header("Access-Control-Allow-Headers", "Origin, X-Requested-With,  
Content-Type, Accept");  
    res.removeHeader('X-Powered-By');  
    if(req.body.id)  
    {  
  
req.body=JSON.parse(modules.crypto.dec(req.body.id.toString(),modules.co  
nfig.salt));  
    }  
    next();  
})  
  
modules.server=modules.http.createServer(modules.app);  
  
module.exports=modules;
```

routes.js

```
module.exports=function(modules){  
    require('./routes/login')(modules);  
    require('./routes/ingredients')(modules);  
    require('./routes/recipes')(modules);  
}
```

services.js

```
module.exports=function(modules) {  
  return {  
    login: require('./services/loginService')(modules),  
    ingredients: require('./services/ingredientService')(modules),  
    recipes: require('./services/recipeService')(modules),  
  }  
}
```

ingredients.js

```
module.exports=function(modules)  
{  
  modules.app.get("/api/getingredients",function(req,res) {  
  
modules.services.ingredients.getIngredients().then(function(data) {  
    res.status(200).send(JSON.stringify({status:true,data:data}));  
  },function(err) {  
    console.log(err)  
  
res.status(400).send(JSON.stringify({status:false,error:err.message}));  
  }).catch(function(err) {  
  
res.status(400).send(JSON.stringify({status:false,error:err.message}));  
  })  
  
  });  
  
  modules.app.post("/register", function(req, res) {  
    modules.services.login.signUp(req.body).then(function(data) {  
      res.status(200).send(JSON.stringify({status:true,data:data}));  
    },function(err) {  
      console.log(err)  
  
res.status(400).send(JSON.stringify({status:false,error:err.message}));  
    }).catch(function(err) {  
  
res.status(400).send(JSON.stringify({status:false,error:err.message}));  
    })  
  })  
}
```

login.js

```
module.exports=function(modules)
{
    modules.app.post("/login",function(req,res) {
        if(_validate(req.body)) {
            modules.services.login.login(req.body).then(function(data) {
                res.status(200).send(JSON.stringify({status:true,data:data}));
            },function(err) {
                console.log(err)
            })
        }
        res.status(400).send(JSON.stringify({status:false,error:err.message}));
    }).catch(function(err) {
        res.status(400).send(JSON.stringify({status:false,error:err.message}));
    })
}
else {
    res.status(200).send(JSON.stringify({status:false,error:error}))
}
});

modules.app.post("/register", function(req, res) {
    modules.services.login.signUp(req.body).then(function(data) {
        res.status(200).send(JSON.stringify({status:true,data:data}));
    },function(err) {
        console.log(err)
    })
    res.status(400).send(JSON.stringify({status:false,error:err.message}));
}).catch(function(err) {
    res.status(400).send(JSON.stringify({status:false,error:err.message}));
})
})

modules.app.get("/test", function(req, res) {
    modules.db.collection('users').find({username:"kailasr"}).toArray(function(err,rows) {
        console.log(err);
        res.send("res", rows)
    })
})
})

function _validate(data)
{

```

```
if(!data.hasOwnProperty('email'))
{
    error="Email Required";
    return false;
}
if(!data.hasOwnProperty('password'))
{
    error="Password Required";
    return false;
}
return true;
}
```

recipes.js

```
module.exports=function(modules)
{
    modules.app.post("/api/getrecipes",function(req,res){
        let ingredients = req.body.ingredients;

        var paramString = ingredients.join(',');

modules.services.recipes.getRecipes(paramString).then(function(data){
    res.status(200).send(JSON.stringify({status:true,data:data}));
},function(err){
    console.log(err);

res.status(400).send(JSON.stringify({status:false,error:err.message}));
}).catch(function(err){

res.status(400).send(JSON.stringify({status:false,error:err.message}));
})

});

modules.app.get("/api/getonerecipe/:id",function(req,res){
    let recipe_id = req.params.id;

modules.services.recipes.getOneRecipe(recipe_id).then(function(data){
    res.status(200).send(JSON.stringify({status:true,data:data}));
},function(err){
    console.log(err);

res.status(400).send(JSON.stringify({status:false,error:err.message}));
}).catch(function(err){
```

```
res.status(400).send(JSON.stringify({status:false,error:err.message}));
    })

    });

    modules.app.get("/api/get",function(req,res) {
        const options = {
            url:
`https://api.spoonacular.com/recipes/716429/information?apiKey=db0ed80b21ee40948f87c62bfa381051&includeNutrition=true`,
            json: true
        }

        modules.rp(options).then((data) => {
            let userData = [];
            modules.fs.writeFileSync("/docs/recipes.js", data);
            res.status(200).send(JSON.stringify({status:true,data:data}));
        }).catch((err) => {
            console.log(err);
        });

        res.status(400).send(JSON.stringify({status:false,error:err.message}));
    });
})
}
```

ingredientService.js

```
module.exports=function(modules) {
    return {
        getIngredients:function () {
            return new Promise((resolve,reject) => {

modules.db.collection('ingredients').find().toArray(function(err,rows) {
                if(err) {
                    reject(err);
                } else {
                    resolve(rows);
                }
            })
        })
    },

}
```

loginService.js

```
module.exports=function(modules) {
```

```
    return {
      login:function (user) {
        return new Promise((resolve,reject) => {
modules.db.collection('users').find({email:user.email}).toArray(function
(err,rows){
          if(err) {
            reject(err);
          } else {
            resolve(rows);
          }
        })
      })
    },
    signUp:function (user) {
      return new Promise((resolve,reject) => {
modules.db.collection('users').insertOne(user,function(err,res){
          if(err) {
            reject(err);
          } else {
            resolve(res);
          }
        })
      })
    }
  }
}
```

recipeService.js

```
module.exports=function(modules) {
  return {
    getRecipes:function (ingredients) {
      return new Promise((resolve,reject) => {

        const options = {
          url:
`https://api.spoonacular.com/recipes/findByIngredients?ingredients=${ing
redients}&number=12`,
          headers: {'x-api-key': 'db0ed80b21ee40948f87c62bfa381051'}
        }

        function callback(error, response, body) {
          if (!error && response.statusCode == 200) {
            const info = JSON.parse(body);
            modules.fs.writeFileSync("rec.txt", body);
          }
        }
      })
    }
  }
}
```

```
        resolve(info);
      } else {
        console.log(error);
        reject(error);
      }
    }
  }

  modules.request(options, callback);

  })
},
getOneRecipe:function (id) {
  return new Promise((resolve,reject) => {
    const options = {
      url:
`https://api.spoonacular.com/recipes/${id}/information`,
      headers: {'x-api-key': 'db0ed80b21ee40948f87c62bfa381051'}
    }

    function callback(error, response, body) {
      if (!error && response.statusCode == 200) {
        const info = JSON.parse(body);
        resolve(info);
      } else {
        reject(error);
      }
    }

    modules.request(options, callback);
  })
},
}
}
```

Angular code:

Index.html

```
<!doctype html>
<html lang="en">
<head>
  <meta charset="utf-8">
  <title>Client</title>
  <base href="/">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <link rel="icon" type="image/x-icon" href="favicon.ico">

  <!-- Bootstrap CSS -->
  <link rel="stylesheet" href="assets/css/bootstrap.min.css">
```



```
<!-- Google Fonts -->
<link
href="https://fonts.googleapis.com/css?family=Poppins:300,300i,400,400i,
600,600i,700,700i|Satisfy|Comic+Neue:300,300i,400,400i,700,700i"
rel="stylesheet">

<!--load all Font Awesome styles -->
<link rel="stylesheet" href="assets/font-awesome/css/all.min.css">

</head>
<body>
  <app-root></app-root>

  <!-- Bootstrap JS -->
  <script src="assets/js/bootstrap.bundle.min.js"></script>

</body>
</html>
```

header.component.html

```
<div class="d-flex flex-column flex-shrink-0 p-3 bg-light">
  <div class="bg">
    <a href="/" class="d-flex align-items-center mb-3 mb-md-0 me-md-auto
link-dark text-decoration-none">
      <svg class="bi me-2" width="40" height="32"><use
xlink:href="#bootstrap"/></svg>
      <span class="fs-4">Pantry</span>
    </a>
    <button (click)="getSelectedIngredients()" class="btn btn-
secondary">Search Recipe</button>
  <hr>
  <ul class="nav nav-pills flex-column mb-auto">
    <li class="nav-item">
      <!-- <a href="#" class="nav-link active" aria-current="page">
        <svg class="bi me-2" width="16" height="16"><use
xlink:href="#home"/></svg>
        Home
      </a> -->
      <form class="col-lg-auto mb-3 mb-lg-0">
        <input type="text" class="form-control" [(ngModel)]="data"
name="data" placeholder="Search..." aria-label="Search">
      </form>
    </li>
    <li>
      <!-- <a href="#" class="nav-link link-dark">
```

```
<svg class="bi me-2" width="16" height="16"><use
xlink:href="#speedometer2"/></svg>

Ingredients
</a> -->

<p *ngFor="let x of selectedIngredients">{{x}}</p>
</li>
<li>
  <!-- <div class="row">
    <div class="col-sm-6"> -->
      <div class="card mt-2">
        <div class="card-body">
          <h5 class="card-title">Vegetables & Greens</h5>
          <p class="card-
text">{{selectedVegies.length}}/{{vegies.length}} Ingredients</p>
          <!-- <p *ngFor="let x of selectedVegies">{{x}}</p> -->
          <span class="gap-3">
            <button *ngFor="let veggy of vegies | slice:0:(more1 ?
undefined : 6) | filter:data" (click)="vegiesClick(veggy)" aria-
pressed="true" class="btn tags-mini-item-desktop active btn-
success">{{veggy}}</button>
          </span>
          <a (click)="more1 = !more1"><i class="fa-solid fa-angles-
{{ more1 ? 'left ' : 'right' }}"></i></a>

        </div>
      </div>
    </div>
  <!-- </div>
  <div class="col-sm-6"> -->
    <div class="card mt-2">
      <div class="card-body">
        <h5 class="card-title">Herbs & Spices</h5>
        <p class="card-
text">{{selectedHerbs.length}}/{{herbs.length}} Ingredients</p>
        <!-- <p *ngFor="let x of selectedHerbs">{{x}}</p> -->
        <span class="gap-3">
          <button *ngFor="let herb of herbs | slice:0:(more2 ?
undefined : 6) | filter:data" (click)="herbsClick(herb)"
[ngClass]="{'tags-mini-item-desktop': toggle, 'selected-tag': !toggle}"
aria-pressed="true" class="btn tags-mini-item-desktop active btn-
outline-success">{{herb}}</button>
          </span>
          <a (click)="more2 = !more2"><i class="fa-solid fa-angles-
{{ more2 ? 'left ' : 'right' }}"></i></a>

        </div>
      </div>

    <div class="card mt-2">
```

```
<div class="card-body">
  <h5 class="card-title">Dairy-Free & Meat Substitutes</h5>
  <p class="card-
text">{{selectedHerbs.length}}/{{substitues.length}} Ingredients</p>
  <!-- <p *ngFor="let x of selectedHerbs">{{x}}</p> -->
  <span class="gap-3">
    <button *ngFor="let substitute of substitues |
slice:0:(more3 ? undefined : 6) | filter:data"
(click)="herbsClick(substitute)" [ngClass]="{'tags-mini-item-desktop':
toggle, 'selected-tag': !toggle}" class="btn tags-mini-item-
desktopactive btn-success">{{substitute}}</button>
  </span>
  <a (click)="more3 = !more3"><i class="fa-solid fa-angles-
{{ more3 ? 'left ' : 'right' }}"></i></a>
</div>
</div>

<div class="card mt-2">
  <div class="card-body">
    <h5 class="card-title">Dairy & Eggs</h5>
    <p class="card-
text">{{selectedHerbs.length}}/{{dairies.length}} Ingredients</p>
    <!-- <p *ngFor="let x of selectedHerbs">{{x}}</p> -->
    <span class="gap-3">
      <button *ngFor="let dairy of dairies | slice:0:(more4 ?
undefined : 6) | filter:data" (click)="herbsClick(dairy)"
[ngClass]="{'tags-mini-item-desktop': toggle, 'selected-tag': !toggle}"
class="btn tags-mini-item-desktop active btn-success">{{dairy}}</button>
      </span>
      <a (click)="more4 = !more4"><i class="fa-solid fa-angles-
{{ more4 ? 'left ' : 'right' }}"></i></a>
    </div>
  </div>

<div class="card mt-2">
  <div class="card-body">
    <h5 class="card-title">Baking</h5>
    <p class="card-
text">{{selectedHerbs.length}}/{{bakings.length}} Ingredients</p>
    <!-- <p *ngFor="let x of selectedHerbs">{{x}}</p> -->
    <span class="gap-3">
      <button *ngFor="let baking of bakings | slice:0:(more5 ?
undefined : 6) | filter:data" (click)="herbsClick(baking)"
[ngClass]="{'tags-mini-item-desktop': toggle, 'selected-tag': !toggle}"
class="btn tags-mini-item-desktop active btn-
success">{{baking}}</button>
      </span>
```

```
        <a (click)="more5 = !more5"><i class="fa-solid fa-angles-
{{ more5 ? 'left' : 'right' }}"></i></a>
    </div>
</div>

<div class="card mt-2">
    <div class="card-body">
        <h5 class="card-title">Sugar & Sweeteners</h5>
        <p class="card-
text">{{selectedHerbs.length}}/{{sugars.length}} Ingredients</p>
        <!-- <p *ngFor="let x of selectedHerbs">{{x}}</p> -->
        <span class="gap-3">
            <button *ngFor="let sugar of sugars | slice:0:(more6 ?
undefined : 6) | filter:data" (click)="herbsClick(sugar)"
[ngClass]='{"tags-mini-item-desktop": toggle, 'selected-tag': !toggle}"
class="btn tags-mini-item-desktop active btn-success">{{sugar}}</button>
        </span>
        <a (click)="more6 = !more6"><i class="fa-solid fa-angles-
{{ more6 ? 'left' : 'right' }}"></i></a>
    </div>
</div>

<div class="card mt-2">
    <div class="card-body">
        <h5 class="card-title">Fruits & Berries</h5>
        <p class="card-
text">{{selectedHerbs.length}}/{{fruits.length}} Ingredients</p>
        <!-- <p *ngFor="let x of selectedHerbs">{{x}}</p> -->
        <span class="gap-3">
            <button *ngFor="let fruit of fruits | slice:0:(more7 ?
undefined : 6) | filter:data" (click)="herbsClick(fruit)"
[ngClass]='{"tags-mini-item-desktop": toggle, 'selected-tag': !toggle}"
class="btn tags-mini-item-desktop active btn-success">{{fruit}}</button>
        </span>
        <a (click)="more7 = !more7"><i class="fa-solid fa-angles-
{{ more7 ? 'left' : 'right' }}"></i></a>
    </div>
</div>

<div class="card mt-2">
    <div class="card-body">
        <h5 class="card-title">Oils & Fats</h5>
        <p class="card-
text">{{selectedHerbs.length}}/{{herbs.length}} Ingredients</p>
        <!-- <p *ngFor="let x of selectedHerbs">{{x}}</p> -->
        <span class="gap-3">
```

```
        <button *ngFor="let oil of oils | slice:0:(more8 ?
undefined : 6) | filter:data" (click)="herbsClick(oil)"
[ngClass]="{'tags-mini-item-desktop': toggle, 'selected-tag': !toggle}"
class="btn tags-mini-item-desktop active btn-success">{{oil}}</button>
        </span>
        <a (click)="more8 = !more8"><i class="fa-solid fa-angles-
{{ more8 ? 'left' : 'right' }}"></i></a>
    </div>
</div>
<!-- </div>
</div> -->
<div class="card mt-2">
    <div class="card-body">
        <h5 class="card-title">Recipes</h5>
        <ul class="dropdown-menu text- small shadow">

        </ul>

    </div>
</div>
</li>
</ul>
<hr>
<div class="dropdown">

</div>
</div>
</div>
```

header.component.css

```
.tags-mini-item-desktop {
    display: inline-block;
    margin: 5px;
    padding: 10px;
    border-radius: 4px;
    background-color: rgba(181,191,200,.2);
    color: rgba(79,80,90,.6);
    font-size: 14px;
    line-height: 20px;
    text-decoration: none;
    cursor: default;
}

.selected-tag {
    background-color: #93c759;
    color: #fff;
}
```

```
.bg {  
    background-color: #45ba5c;  
}
```

header.component.ts

```
import { Component, OnInit } from '@angular/core';  
import { Router } from '@angular/router';  
import { IngredientService } from '../services/ingredient.service';  
import { ShareIngredientsService } from '../services/share-  
ingredients.service';  
  
@Component({  
    selector: 'app-header',  
    templateUrl: './header.component.html',  
    styleUrls: ['./header.component.css']  
})  
export class HeaderComponent implements OnInit {  
  
    more1: boolean=false;  
    more2: boolean=false;  
    more3: boolean=false;  
    more4: boolean=false;  
    more5: boolean=false;  
    more6: boolean=false;  
    more7: boolean=false;  
    more8: boolean=false;  
    vegies: string[] = [];  
    herbs: string[] = [];  
    dairies: string[]=[];  
    bakings: string[]=[];  
    sugars: string[]=[];  
    fruits: string[]=[];  
    oils: string[]=[];  
    substitues: string[]=[];  
    selectedVegies: string[] = [];  
    selectedHerbs: string[] = [];  
    selectedDairies: string[] = [];  
    selectedBakings: string[]=[];  
    selectedSugars: string[] = [];  
    selectedFruits: string[] = [];  
    selectedOils: string[] = [];  
    selectedSubstitues: string[] = [];  
    selectedIngredients: string[]=[];  
    data: string = '';  
    toggle: boolean = true;
```

```
constructor(private ingredient: IngredientService, private shared:
ShareIngredientsService, private router: Router) { }

ngOnInit(): void {
    // this.vegies = ['garlic', 'onion', 'carrot', 'bell pepper',
'scalian']
    this.getIngredients();
    this.shared.ingrArr.subscribe(ingrArr => this.selectedIngredients =
ingrArr)
}

getSelectedIngredients() {
    this.shared.getSelectedIngredients(this.selectedIngredients)
    this.router.navigate(['/recipes']);
}

getIngredients() {
    this.ingredient.getIngredients().subscribe(res => {
        const body = JSON.parse(JSON.stringify(res));
        console.log(body);
        if(body.status == true) {
            this.vegies = body.data.filter((ele: any) => ele.category ==
"Vegetables and greens")[0].ingredients
            this.herbs = body.data.filter((ele: any) => ele.category ==
"Herbs and spices")[0].ingredients;
            this.substitues = body.data.filter((ele: any) => ele.category ==
"Dairy-Free & Meat Substitutes")[0].ingredients;
            this.dairies = body.data.filter((ele: any) => ele.category ==
"Dairy & Eggs")[0].ingredients;
            this.bakings = body.data.filter((ele: any) => ele.category ==
"Baking")[0].ingredients;
            this.sugars = body.data.filter((ele: any) => ele.category ==
"Sugar & Sweeteners")[0].ingredients;
            this.fruits = body.data.filter((ele: any) => ele.category ==
"Fruits & Berries")[0].ingredients;
            this.oils = body.data.filter((ele: any) => ele.category == "Oils
& Fats")[0].ingredients;

        } else {
            console.log(body.data);
        }
    })
}

herbsClick(herb: string) {
    this.getAllSelectedIngredients(herb)
    if(this.selectedHerbs.includes(herb)) {
```

```
        this.selectedHerbs = this.selectedHerbs.filter(item => item !==
herb);
    } else {
        this.selectedHerbs.push(herb);
    }

    // this.value = (!this.toggle)?"View":"Hide";
    // this.toggle = !this.toggle;
}

vegiesClick(veggy: string) {
    this.getAllSelectedIngredients(veggy);
    if(this.selectedVegies.includes(veggy)) {
        this.selectedVegies = this.selectedVegies.filter(item => item !==
veggy);
    } else {
        this.selectedVegies.push(veggy);
    }

    // this.value = (!this.toggle)?"View":"Hide";
    // this.toggle = !this.toggle;
}

dairiesClick(veggy: string) {
    this.getAllSelectedIngredients(veggy);
    if(this.selectedDairies.includes(veggy)) {
        this.selectedDairies = this.selectedDairies.filter(item =>
item !== veggy);
    } else {
        this.selectedDairies.push(veggy);
    }

    // this.value = (!this.toggle)?"View":"Hide";
    // this.toggle = !this.toggle;
}

bakingsClick(veggy: string) {
    this.getAllSelectedIngredients(veggy);
    if(this.selectedBakings.includes(veggy)) {
        this.selectedBakings = this.selectedBakings.filter(item =>
item !== veggy);
    } else {
        this.selectedBakings.push(veggy);
    }

    // this.value = (!this.toggle)?"View":"Hide";
    // this.toggle = !this.toggle;
}
```



```
sugarsClick(veggy: string) {
    this.getAllSelectedIngredients(veggy);
    if(this.selectedSugars.includes(veggy)) {
        this.selectedSugars = this.selectedSugars.filter(item => item !==
veggy);
    } else {
        this.selectedSugars.push(veggy);
    }

    // this.value = (!this.toggle)?"View":"Hide";
    // this.toggle = !this.toggle;
}

fruitsClick(veggy: string) {
    this.getAllSelectedIngredients(veggy);
    if(this.selectedFruits.includes(veggy)) {
        this.selectedFruits = this.selectedFruits.filter(item => item !==
veggy);
    } else {
        this.selectedFruits.push(veggy);
    }

    // this.value = (!this.toggle)?"View":"Hide";
    // this.toggle = !this.toggle;
}

oilsClick(veggy: string) {
    this.getAllSelectedIngredients(veggy);
    if(this.selectedOils.includes(veggy)) {
        this.selectedOils = this.selectedOils.filter(item => item !==
veggy);
    } else {
        this.selectedVegies.push(veggy);
    }

    // this.value = (!this.toggle)?"View":"Hide";
    // this.toggle = !this.toggle;
}

getAllSelectedIngredients(ingredient: string) {
    if(this.selectedIngredients.includes(ingredient)) {
        this.selectedIngredients = this.selectedIngredients.filter(item =>
item !== ingredient);
    } else {
        this.selectedIngredients.push(ingredient);
    }
}
```

```
}
```

home.component.html

```
<router-outlet></router-outlet>
```

home.component.ts

```
import { Component, OnInit } from '@angular/core';

@Component({
  selector: 'app-home',
  templateUrl: './home.component.html',
  styleUrls: ['./home.component.css']
})
export class HomeComponent implements OnInit {

  constructor() { }

  ngOnInit(): void {
  }

}
```

footer.component.html

```
<main class="ms-sm-auto px-md-3">
  <div class="d-flex justify-content-between flex-wrap flex-md-nowrap
align-items-center pt-3 pb-2 mb-3 border-bottom">
    <h1 class="h2" routerLink="/">
      <img src = "../assets/img/plate.png">
    </h1>
    <div class="btn-toolbar mb-2 mb-md-0">
      <form class="col-12 col-lg-auto mb-3 mb-lg-0 me-lg-3">
        <input type="search" class="form-control"
placeholder="Search..." aria-label="Search">
      </form>
      <!-- <button type="button" class="btn btn-sm btn-outline-
secondary"> -->
      <!-- <span data-feather="calendar"><i class="fa-solid fa-user-
check"></i><a class="btn btn-sm btn-outline-secondary"
routerLink="/login"></a></span> -->
      <a routerLink="/login" class="nav-link padding-y-xs padding-x-
md" aria-label="account">
        <i class="fa-solid fa-user"></i>
      </a>
      <!-- Login -->
      <!-- </button> -->
```

```
</div>
</div>
<div [hidden]="hideElement">
  <img src = "../assets/img/back.jpg" height="100%" width="100%">
</div>
</main>
```

footer.component.ts

```
import { Component, OnInit } from '@angular/core';
import { RouterModule, Router, NavigationEnd } from '@angular/router';

@Component({
  selector: 'app-footer',
  templateUrl: './footer.component.html',
  styleUrls: ['./footer.component.css']
})
export class FooterComponent implements OnInit {

  hideElement = false;
  constructor(private router: Router) {
    this.router.events.subscribe((event) => {
      if (event instanceof NavigationEnd) {
        if (event.url === '/login' || event.url === '/register' || event.url
=== '/recipes' || event.url === '/recipe/:id') {
          this.hideElement = true;
        } else {
          this.hideElement = false;
        }
      }
    });
  }

  ngOnInit(): void {
  }
}
```

login.component.html

```
<div class="text-center">
  <div class="bg-image d-flex justify-content-center align-items-
center" style="background-image: url('assets/img/sign.png'); height:
100vh;">
    <!-- <h1 class="text-white">Find Your Recipe</h1> -->
    <div class="col-6">
      <form [formGroup]="user" (submit)="submit()">
        <!-- <h1 class="h3 mb-3 fw-normal">Please sign in</h1> -->
```

```
<img src = "../assets/img/plate.png">
<div class="form-floating">
  <input type="email" class="form-control"
formControlName="email" id="floatingInput"
placeholder="name@example.com">
  <label for="floatingInput">Email</label>
</div>
<div class="form-floating">
  <input type="password" class="form-control"
formControlName="password" id="floatingPassword" placeholder="Password">
  <label for="floatingPassword">Password</label>
</div>

<!-- <div class="checkbox mb-3">
  <label>
    <input type="checkbox" value="remember-me"> Remember me
  </label>
</div> -->
<button class="w-100 btn btn-secondary btn-lg btn-block"
type="submit">Log in</button>
<!-- <p class="mt-5 mb-3 text-muted">&copy; 2017-2021</p> -->
</form>
<a class="router-link mt-3" routerLink="/register">Sign Up</a>
</div>

</div>
</div>
```

Login.component.ts

```
import { Component, OnInit } from '@angular/core';
import { FormBuilder, FormGroup, Validators } from '@angular/forms';
import { Router } from '@angular/router';
import { LoginService } from '../services/login.service';

@Component({
  selector: 'app-login',
  templateUrl: './login.component.html',
  styleUrls: ['./login.component.css']
})
export class LoginComponent implements OnInit {

  user: FormGroup;
  result: any;

  constructor(private formBuilder: FormBuilder, private login:
LoginService, private router: Router) {
```

```
this.user = this.formBuilder.group({
  email: ['', Validators.required],
  password: ['', Validators.required]
})

}

ngOnInit(): void {

}

submit() {
  if (this.user.valid) {
    this.login.login(this.user.value).subscribe(res => {
      console.log(res)
      const body = JSON.parse(JSON.stringify(res));
      if (body.status == true) {
        alert("Welcome to Delicious!");
        this.router.navigate(['/']);
      } else {
        console.log(body.data);
        this.result = body.data;
      }
    })
  }
}

}
```

recipe.component.html

```
<div class="card mb-3">
  
  <div class="card-body">
    <h5 class="card-title">{{title}}</h5>
    <p class="card-text" [innerHTML]="summary"></p>
    <p class="card-text"><b>Instructions</b><small class="text-muted">{{instructions}}</small></p>
    <a href="{{sourceUrl}}" target="_blank" class="btn btn-primary">Visit Recipe</a>
  </div>
</div>
```

recipe.component.ts

```
import { Component, OnInit } from '@angular/core';
import { DomSanitizer, SafeHtml } from '@angular/platform-browser';
```

```
import { ActivatedRoute } from '@angular/router';
import { RecipeService } from '../services/recipe.service';
import { RouterModule, Router, NavigationEnd } from '@angular/router';
import { FooterComponent } from '../footer/footer.component';

@Component({
  selector: 'app-recipe',
  templateUrl: './recipe.component.html',
  styleUrls: ['./recipe.component.css']
})
export class RecipeComponent implements OnInit {

  hideElement = false;

  recipeObj: any;
  imgSrc: string = '';
  instructions: string = '';
  readyInMinutes: string = '';
  sourceUrl: string = '';
  summary: SafeHtml = '';
  title: string = '';
  vegetarian: boolean = false;

  constructor(private route: ActivatedRoute, private sanitizer:
DomSanitizer, private recipe: RecipeService, private router: Router) {
    this.router.events.subscribe((event) => {
      if (event instanceof NavigationEnd) {
        if (event.url === '/login' || event.url === '/register' || event.url
=== '/recipes' || event.url === '/recipe/:id') {
          this.hideElement = true;
        } else {
          this.hideElement = false;
        }
      }
    });
  }

  ngOnInit(): void {
    this.route.params.subscribe(params => {
      this.getRecipe(params['id']);
    });
  }

  getRecipe(recipe_id: string) {
    console.log(recipe_id)
    this.recipe.getOneRecipe(recipe_id).subscribe(res => {
      const body = JSON.parse(JSON.stringify(res));
    });
  }
}
```

```
    console.log(body);
    if (body.status == true) {
        this.title = body.data.title;
        this.imgSrc = body.data.image;
        this.summary =
this.sanitizer.bypassSecurityTrustHtml (body.data.summary);
        this.sourceUrl = body.data.sourceUrl;
        this.instructions = body.data.instructions;
        this.vegetarian = body.data.vegetarian;
    } else {
        console.log(body.data);
    }
})
}
```

recipes.component.html

```
<div class="row">
  <div class="col-sm-4 mt-2" *ngFor="let recipe of recipes">
    <div class="card">
      
      <div class="card-body">
        <h5 class="card-title">{{recipe.title}}</h5>

        <div class="container px-4">
          <div class="row gx-5">
            <div class="col">
              <div class="p-3 border bg-light">You have: </div>
              <li *ngFor="let x of recipe.usedIngredients">{{x.name}}</li>
            </div>
            <div class="col">
              <div class="p-3 border bg-light">You're Missing: </div>
              <li *ngFor="let x of recipe.missedIngredients">{{x.name}}</li>
            </div>
          </div>
        </div>

        <a class="btn btn-primary" (click)="visit(recipe.id)">Visit Recipe</a>
        <!-- <a [routerLink]="['recipe', recipe.id]" target="_blank" class="btn btn-primary">Visit
Recipe</a> -->
      </div>
    </div>
  </div>
</div>
```

recipes.component.ts

```
import { Component, Input, OnInit } from '@angular/core';
import { DomSanitizer, SafeHtml } from '@angular/platform-browser';
import { Router } from '@angular/router';
import { IngredientService } from '../services/ingredient.service';
```

```
import { ShareIngredientsService } from '../services/share-ingredients.service';

@Component({
  selector: 'app-recipes',
  templateUrl: './recipes.component.html',
  styleUrls: ['./recipes.component.css']
})
export class RecipesComponent implements OnInit {

  recipes: any;
  ingrArr: string[][] = [];
  title: string = '';
  summary: SafeHtml = '';
  img: string = '';
  prep_time: string = '';
  servings: string = '';
  sourceUrl: string = '';

  constructor(private router: Router, private ingredient: IngredientService, private shared: ShareIngredientsService, private sanitizer: DomSanitizer) { }

  @Input() recipeArr: string[] = [];

  ngOnInit(): void {
    console.log(this.ingrArr)
    this.shared.ingrArr.subscribe(ingrArr => this.ingrArr = JSON.parse(JSON.stringify(ingrArr)))
    this.getRecipes(this.ingrArr)
  }

  getRecipes(ingredients: string[]) {
    console.log(ingredients);
    this.ingredient.getRecipes(ingredients).subscribe(res => {
      const body = JSON.parse(JSON.stringify(res));
      // console.log(body);
      if (body.status == true) {
        this.recipes = body.data;
        this.title = body.data.title;
        this.summary =
this.sanitizer.bypassSecurityTrustHtml(body.data.summary);
        this.img = body.data.image;
        this.prep_time = body.data.readyInMinutes;
        this.servings = body.data.servings;
        this.sourceUrl = body.data.sourceUrl
        console.log(this.recipes);
      } else {
```



```
        console.log(body.data);
    }
    })
}

visit(id: string) {
    this.router.navigate(['recipe', id]);
}

}
```

Register.component.html

```
<form class="needs-validation" [formGroup]="user" (ngSubmit)="submit()"
novalidate>
    <div class="row g-3">
        <div class="col-sm-6">
            <label for="firstName" class="form-label">First name</label>
            <input type="text" class="form-control" id="firstName"
formControlName="firstName" placeholder="" value="" required>
            <div class="invalid-feedback">
                Valid first name is required.
            </div>
        </div>

        <div class="col-sm-6">
            <label for="lastName" class="form-label">Last name</label>
            <input type="text" class="form-control" id="lastName"
formControlName="lastName" placeholder="" value="" required>
            <div class="invalid-feedback">
                Valid last name is required.
            </div>
        </div>

        <div class="col-sm-6">
            <label for="email" class="form-label">Email</label>
            <input type="email" class="form-control" id="email"
formControlName="email" placeholder="you@example.com">
            <div class="invalid-feedback">
                Please enter a valid email address for shipping updates.
            </div>
        </div>

        <div class="col-sm-6">
            <label for="password" class="form-label">Password</label>
            <input type="password" class="form-control" id="password"
formControlName="password" placeholder="*****">
            <div class="invalid-feedback">
```

```
        Please enter a strong password.
    </div>
</div>

<button class="btn btn-primary" type="submit">Sign-Up</button>
</div>
</form>
```

register.component.ts

```
import { Component, OnInit } from '@angular/core';
import { FormBuilder, FormGroup, Validators } from '@angular/forms';
import { Router } from '@angular/router';
import { LoginService } from '../services/login.service';

@Component({
  selector: 'app-register',
  templateUrl: './register.component.html',
  styleUrls: ['./register.component.css']
})
export class RegisterComponent implements OnInit {

  // user: any;
  result: any;
  user: FormGroup;

  constructor(private login: LoginService, private formBuilder:
FormBuilder, private router: Router) {

    this.user = this.formBuilder.group({
      firstName: ['', Validators.required],
      lastName: ['', Validators.required],
      email: ['', Validators.required],
      password: ['', Validators.required]
    })
  }

  ngOnInit(): void {

  }

  submit() {
    console.log(this.user.value)
    if (this.user.valid) {
      this.login.register(this.user.value).subscribe(res => {
        const body = JSON.parse(JSON.stringify(res));
        if (body.status == true) {
          alert("Welcome to Delicious!");
        }
      });
    }
  }
}
```

```
        this.router.navigate(['/']);
    } else {
        console.log(body.data);
        this.result = body.data;
    }
    })
}
}
```

ingredient.service.spec

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

import { IngredientService } from './ingredient.service';

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

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

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

ingredient.service

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

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

    url: string = 'http://localhost:3000/api/';

    constructor(private http: HttpClient) { }

    getIngredients() {
        return this.http.get(this.url+'getingredients');
    }
}
```

```
}

    getRecipes(ingredients: string[]) {
        return this.http.post(this.url+'getrecipes', {ingredients:
ingredients}, {headers: new HttpHeaders({'content-type':
'application/json'}}));
    }
}
```

login.service.spec

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

import { LoginService } from './login.service';

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

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

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

login.service

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

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

    url: string = "http://localhost:3000/"
    constructor(private http: HttpClient) { }

    login(user: any) {
        return this.http.post(this.url+"login", user, {headers: new
HttpHeaders({'content-type': 'application/json'}}));
    }
}
```

```
    register(user: any) {  
        return this.http.post(this.url+"register", user, {headers: new  
HttpHeaders({'content-type': 'application/json'}}));  
    }  
}
```

recipe.service.spec

```
import { TestBed } from '@angular/core/testing';  
  
import { RecipeService } from './recipe.service';  
  
describe('RecipeService', () => {  
    let service: RecipeService;  
  
    beforeEach(() => {  
        TestBed.configureTestingModule({});  
        service = TestBed.inject(RecipeService);  
    });  
  
    it('should be created', () => {  
        expect(service).toBeTruthy();  
    });  
});
```

recipe.service

```
import { HttpClient } from '@angular/common/http';  
import { Injectable } from '@angular/core';  
  
@Injectable({  
    providedIn: 'root'  
})  
export class RecipeService {  
  
    url: string = 'http://localhost:3000/api/';  
  
    constructor(private http: HttpClient) { }  
  
    getOneRecipe(id: string) {  
        return this.http.get(this.url+'getonerecipe/'+id);  
    }  
}
```

register.service.spec

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

import { RegisterService } from '../register.service';

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

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

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

register.service

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

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

  constructor() { }
}
```

share-ingredient.service.spec

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

import { ShareIngredientsService } from '../share-ingredients.service';

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

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

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

```
});  
});
```

share-ingredient.service

```
import { Injectable } from '@angular/core';  
import { BehaviorSubject } from 'rxjs';  
  
@Injectable({  
  providedIn: 'root'  
})  
export class ShareIngredientsService {  
  
  private ingredientSource = new BehaviorSubject(['']); // set default  
status  
  ingrArr = this.ingredientSource.asObservable();  
  
  constructor() { }  
  
  getSelectedIngredients(ingrArr: string[]) {  
    this.ingredientSource.next(ingrArr)  
  }  
  
}
```

app-routing.module.ts

```
import { NgModule } from '@angular/core';  
import { RouterModule, Routes } from '@angular/router';  
import { HomeComponent } from '../home/home.component';  
import { LoginComponent } from '../login/login.component';  
import { RecipeComponent } from '../recipe/recipe.component';  
import { RecipesComponent } from '../recipes/recipes.component';  
import { RegisterComponent } from '../register/register.component';  
  
const routes: Routes = [  
  {path: "home", component: HomeComponent},  
  {path: "recipe/:id", component: RecipeComponent},  
  {path: "recipes", component: RecipesComponent},  
  {path: "login", component: LoginComponent},  
  {path: "register", component: RegisterComponent},  
];  
  
@NgModule({  
  imports: [RouterModule.forRoot(routes)],  
  exports: [RouterModule]  
})  
export class AppRoutingModule { }
```

app.component.html

```
<div class="container-fluid">
  <div class="row">
    <div class="col-md-3">
      <app-header></app-header>
    </div>
    <div class="col-md-9">
      <app-footer></app-footer>
      <app-home></app-home>
    </div>
  </div>
</div>
```

app.component.ts

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

@Component({
  selector: 'app-root',
  templateUrl: './app.component.html',
  styleUrls: ['./app.component.css']
})
export class AppComponent {
  title = 'client';
}
```

app.module.ts

```
import { NgModule } from '@angular/core';
import { BrowserModule } from '@angular/platform-browser';

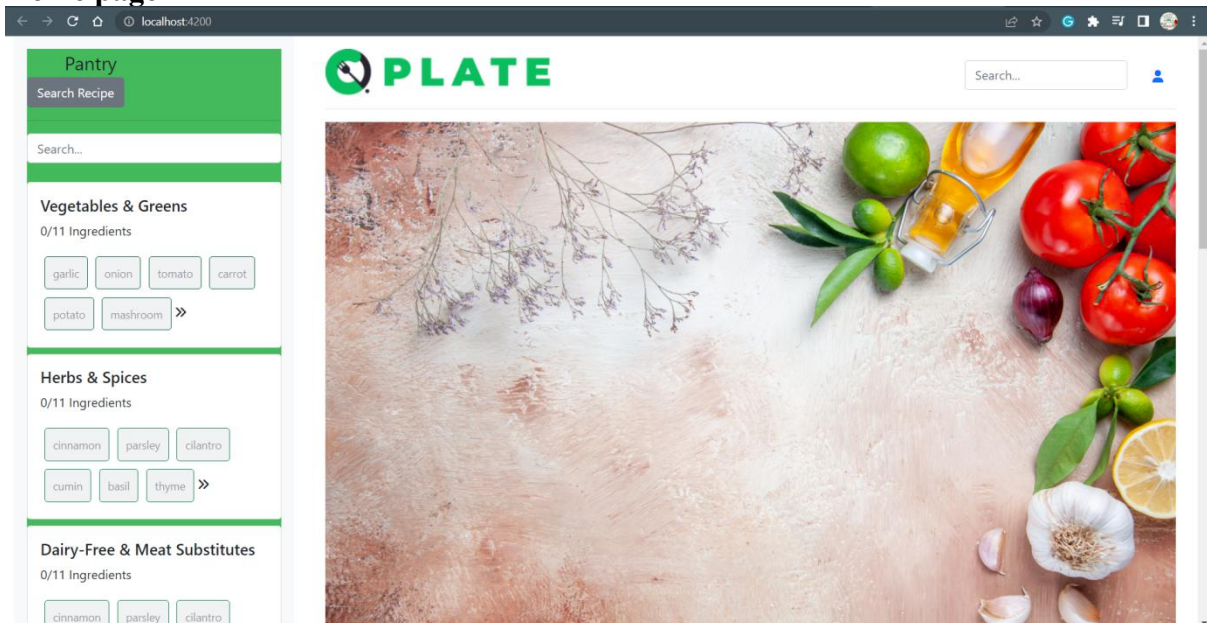
import { AppRoutingModule } from './app-routing.module';
import { AppComponent } from './app.component';
import { HomeComponent } from './home/home.component';
import { LoginComponent } from './login/login.component';
import { RegisterComponent } from './register/register.component';
import { FooterComponent } from './footer/footer.component';
import { HeaderComponent } from './header/header.component';
import { FormsModule, ReactiveFormsModule } from '@angular/forms';
import { HttpClientModule } from '@angular/common/http';
import { Ng2SearchPipeModule } from 'ng2-search-filter';
import { RecipesComponent } from './recipes/recipes.component';
import { RecipeComponent } from './recipe/recipe.component';

@NgModule({
  imports: [
    BrowserModule,
    AppRoutingModule,
    FormsModule,
    ReactiveFormsModule,
    HttpClientModule,
    Ng2SearchPipeModule,
    RecipesComponent,
    RecipeComponent
  ],
  declarations: [
    AppComponent,
    HomeComponent,
    LoginComponent,
    RegisterComponent,
    FooterComponent,
    HeaderComponent
  ],
  providers: [],
  bootstrap: [AppComponent]
})
export class AppModule {}
```

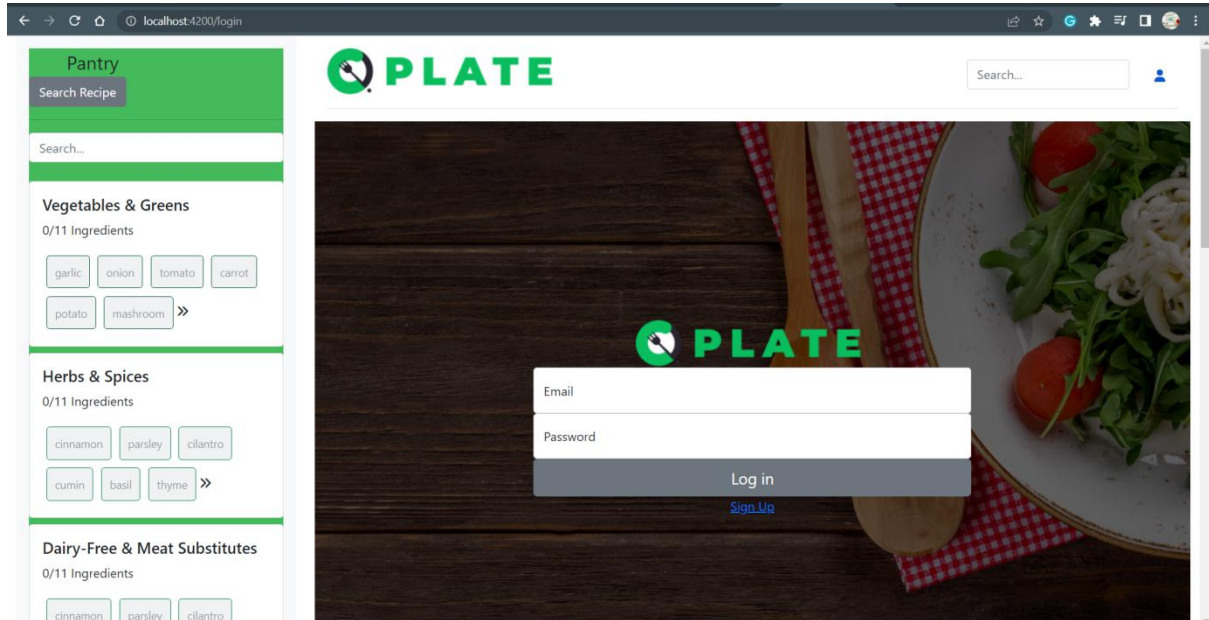


```
declarations: [  
  AppComponent,  
  HomeComponent,  
  LoginComponent,  
  RegisterComponent,  
  FooterComponent,  
  HeaderComponent,  
  RecipesComponent,  
  RecipeComponent  
,  
imports: [  
  BrowserModule,  
  AppRoutingModule,  
  FormsModule,  
  ReactiveFormsModule,  
  HttpClientModule,  
  Ng2SearchPipeModule  
,  
providers: [],  
bootstrap: [AppComponent]  
})  
export class AppModule { }
```

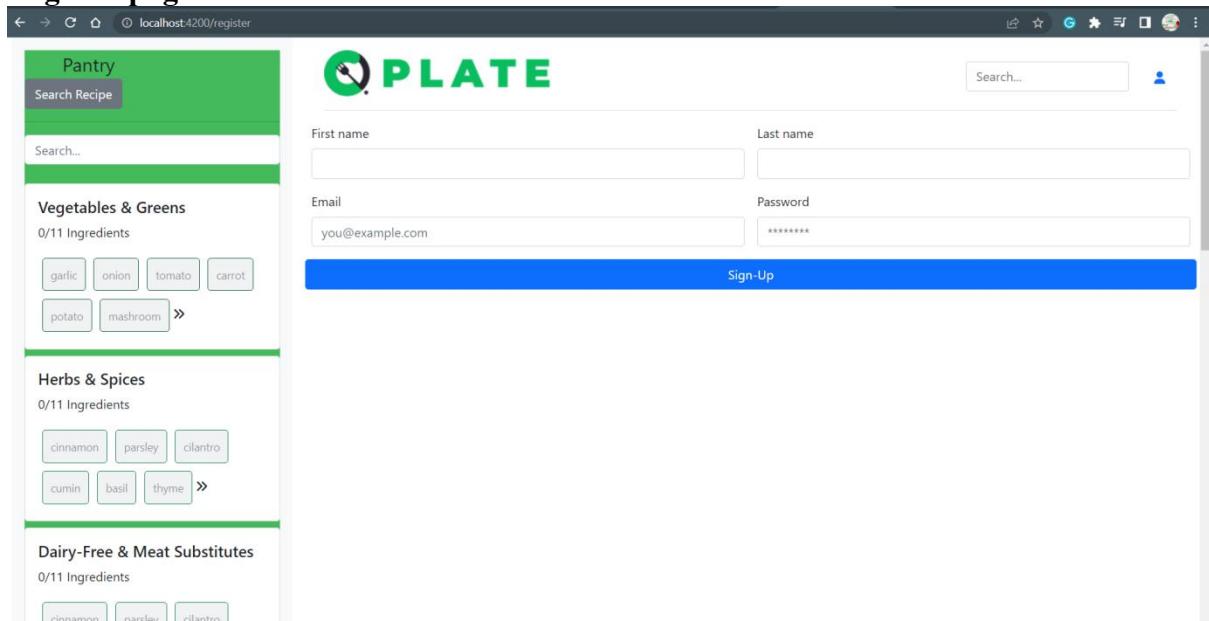
5.5 Snapshots of UI Home page



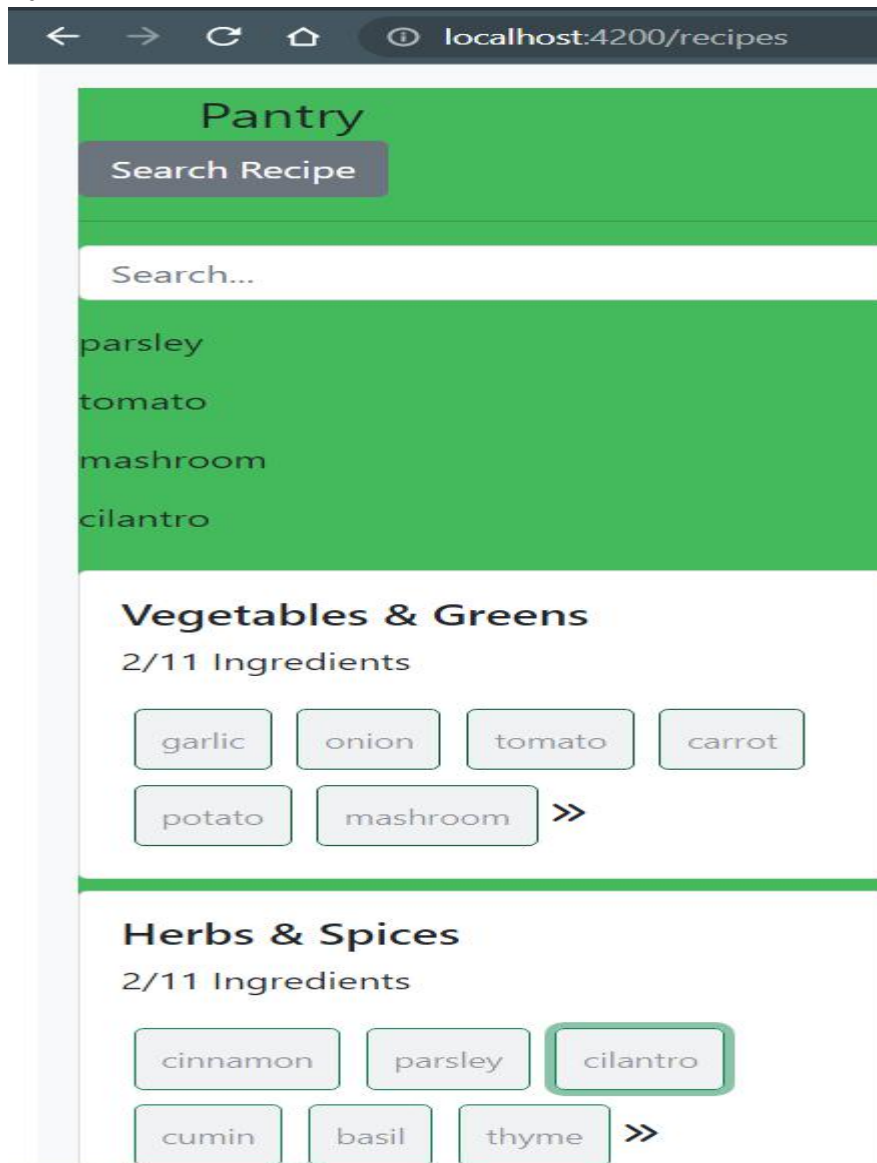
login



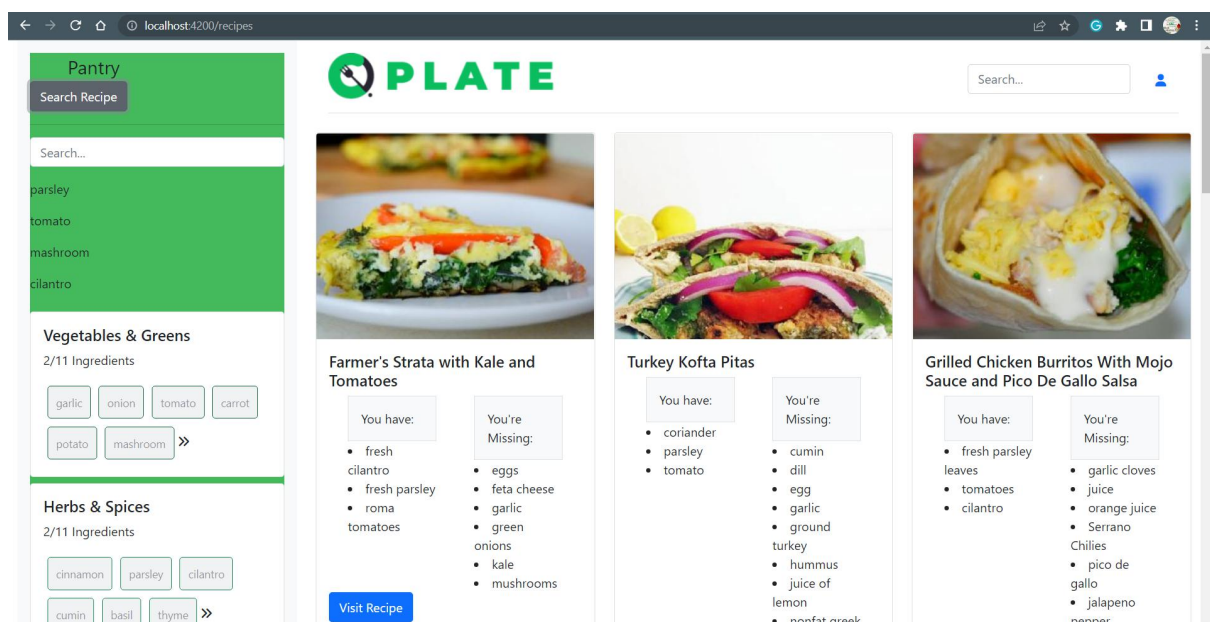
Register page



Ingredients selected



After clicking on search recipe button



cumin

basil

thyme

>>

Dairy-Free & Meat Substitutes

2/11 Ingredients

cinnamon

parsley

cilantro

cumin

basil

thyme

>>

Dairy & Eggs

2/38 Ingredients

Butter

egg

milk

sour cream

heavy cream

whipped cream

>>

Baking

2/43 Ingredients

flour

vanilla

baking powder

baking soda

corn starch

yeast

>>

Visit Recipe


lemon

- nonfat greek yogurt
- pitas
- red onion
- turmeric
- yellow onion

Visit Recipe

- jalapeno pepper
- white onion
- bell pepper
- chili powder
- flour tortillas
- whole chicken
- cheese
- chile garlic sauce

Visit Recipe




Albondigas (Meatball Soup)

You have:

- cilantro
- fresh parsley

You're Missing:




Moroccan Chicken Pitas

You have:

- coriander
- parsley

You're Missing:

- pita bread



Tomato-Pomegranate Jam

You have:

- canned tomatoes

You're Missing:

Lemon

lime

apple

orange

banana

strawberry

>>

Oils & Fats

2/11 Ingredients

Olive oil

vegetable oil

extra virgin olive oil

Canola oil

coconut oil

sesame oil

>>

Recipes

flakes

- white rice

Visit Recipe


greek yogurt

cucumber

mint

bread bowl

Visit Recipe



Sun Dried Tomato and Herb Baked Eggs


You have:

- fresh parsley
- sundried tomatoes

You're Missing:

- eggs
- fresh basil

Visit Recipe



Summer Beet Side Salad


You have:

- parsley
- tomato

You're Missing:

- beets
- english cucumber

Visit Recipe



Homemade Guacamole

You have:

- cilantro
- roma tomato

You're Missing:

- avocados
- juice of lime
- red onion

Visit Recipe

Selected on of the recipe from the list

cinnamon

parsley

cilantro

cumin

basil

thyme

>>

Dairy & Eggs

2/38 Ingredients

Butter

egg

milk

sour cream

heavy cream

whipped cream

>>

Baking

2/43 Ingredients

flour

vanilla

baking powder

baking soda

corn starch

yeast

>>


Sugar & Sweeteners

2/20 Ingredients

Sugar

brown sugar

honey



Meghna Wankhede(2020510068)

52

Sugarbrown sugarhoney
maple syrupcorn syrup
coconut sugar»

Fruits & Berries
2/27 Ingredients
Lemonlimeappleorange
bananastrawberry»

Oils & Fats
2/11 Ingredients

Farmer's Strata with Kale and Tomatoes

Farmer's Strata with Kale and Tomatoes might be just the main course you are searching for. Watching your figure? This gluten free, primal, and vegetarian recipe has **223 calories**, **18g of protein**, and **12g of fat** per serving. This recipe serves 4 and costs \$1.67 per serving. Head to the store and pick up kale, eggs, pepper, and a few other things to make it today. To use up the eggs you could follow this main course with the [Rose Levy Beranbaum's Chocolate Tomato Cake with Mystery Ganache](#) as a dessert. Only a few people made this recipe, and 3 would say it hit the spot. All things considered, we decided this recipe **deserves a spoonacular score of 73%**. This score is solid. Try [Farmer's Strata](#), [Farmer's Strata](#), and [Kale Strata](#) for similar recipes.

Instructions Preheat oven to 325 degrees F.Beat eggs, salt, and pepper, in a medium bowl. Set aside.Heat oil in a 10" oven safe skillet over medium heat.Add mushrooms, green onions, and sliced kale stems to skillet. Sauté for 4 minutes.Add garlic and herbs. Sauté for an additional 2 minutes. Herbs will be wilted and the mushrooms are starting to brown. Handful of fresh parsley, chopped (about 3 Tablespoons)Add layer of kale to mixture. Do not mix in, just lay on top of mushroom and herb mixture.Add a layer of tomatoes over kale and sprinkle crumbled feta over tomatoes.Evenly pour eggs over tomatoes and feta. Cook for 5 minutes over medium heat.As eggs are cooking, lift edges of strata with a spatula allowing eggs in fill in the gaps.Once bottom of strata begins to brown, transfer to the oven. Cook for 20 minutes, or until eggs are set.Remove from pan and cut into wedges.Remove from oven and take strata out of pan with a large spatula. Slice into wedges and serve.

[Visit Recipe](#)

After clicking on the vist recipe you will be redirected to the recipe blog

← → ×

foodista.com/recipe/4ZKH33CK/farmer-s-strata-with-kale-and-tomatoes

☆

🔍

📄

🔗

👤


⋮

[Add a Recipe](#)[Sign In](#)[Sign Up](#)

Foodista

[Home](#)[Blog](#)[Recipes & Cooking](#)[Food News](#)[Holidays](#)[Health & Nutrition](#)[Video & Podcasts](#)

Farmer's Strata with Kale and Tomatoes



[Tweet](#)[Pin It](#)

- 6 Add layer of kale to mixture. Do not mix in, just lay on top of mushroom and herb mixture.
- 7 Add a layer of tomatoes over kale and sprinkle crumbled feta over tomatoes.
- 8 Evenly pour eggs over tomatoes and feta. Cook for 5 minutes over medium heat.
- 9 As eggs are cooking, lift edges of strata with a spatula allowing eggs in fill in the gaps.
- 10 Once bottom of strata begins to brown, transfer to the oven. Cook for 20 minutes, or until eggs are set.
- 11 Remove from pan and cut into wedges.
- 12 Remove from oven and take strata out of pan with a large spatula. Slice into wedges and serve.

0 Comments

Sort by [Oldest](#)

Add a comment...

Ingredients

8 eggs
1/4 teaspoon salt
1/8 teaspoon pepper
1 teaspoon olive oil or coconut oil
1 cup sliced Shitake mushrooms, about 4-5
2 green onions, white and green segments sliced
2 cloves of garlic, minced
Handful of fresh parsley, chopped (about 3 Tablespoons)
Handful of fresh cilantro, chopped (about 3 Tablespoons)
6 large leaves of red kale, chopped and stems removed and sliced
4 Roma tomatoes, sliced into 1/4" slices
1/4 cup Feta cheese, crumbled

Preparation

- 1 Preheat oven to 325 degrees F.
- 2 Beat eggs, salt, and pepper, in a medium bowl. Set aside.
- 3 Heat oil in a 10" oven safe skillet over medium heat.
- 4 Add mushrooms, green onions, and sliced kale stems to skillet. Saute for 4 minutes.
- 5 Add garlic and herbs. Saute for an additional 2 minutes. Herbs will

A Safety Ring



A symbol of your promise
to protect children in India



Related Cooking Videos

[BOUXBEE COOKING SCHOOL & COOKING CLASSES](#)

5.6 Test Cases

Test Case ID	Test Case Name	Test Data	Expected Output	Actual Output	Result
1	User enters username and password	Enters the correct username and password	Logged in successfully.	Home page	Pass
2	User enters username and password	Enters the wrong username and password	Prompt error	Prompt error	Pass

Table no. 4.2.1 (For SignUp)

Test Case ID	Test Case Name	Test Data	Expected Output	Actual Output	Result
1	User enters username and password	Valid username and password which doesn't exist in database	Registered successfully.	Login page	Pass
2	User enters username and password	Invalid username and password which contains in database.	Prompt error	Prompt error	Pass

Table no. 4.2.2 (For Login)

Chapter 6

Documentation & Installation

Step 1: Download NodeJS and Angular

Step 2: In the command prompt, execute the following command:

```
npm install -g @angular/cli
```

Step 3: Set the environment and then run the following command in the terminal:

```
npm install  
node index.js
```


Chapter 7

Future Enhancements

This project is developed as a master's project and still gives lot of scope for its extension if it is going to be developed as commercial product. Below discussed are some of the features that can be added to the application.

- More filter options can be provided for the users to filter the search results such as filtering the results according to cuisine type, ingredients type.
- More sorting options can also be added like sorting results according to nutritional elements in recipe.
- Search options according to taste type can also be added, for example search recipes only with more salty taste.
- Users can share their favorite collection to others users on social networking platforms.
- Users can add recipes to the web application.
- Adding admin side module to validate the recipes added by users.
- Providing personalized recommendations to users by tracking their search history and liked recipes.

Chapter 8

Limitations

- All the users should have active internet connection.
- The supportive web browser is needed.

Chapter 9

Conclusion

This Recipe Search Engine web application is developed to work as a central information hub for the kitchen—connecting consumers with recipe ideas, ingredient lists, and cooking instructions. The requirements are collected by thinking what all features the user expects to present in a food blog like application. Angular is used to design web pages and implement MVC architecture. Bootstrap, js are used for creating interactive user interface. RESTful web services is used to communicate with the spoonacular API and pull data from it. Mongodb is used as database engine and entity framework is used to connect application with the database. The performance of this application is evaluated by rigorously testing it against various test scenarios. Efficiency and correctness of the application is evaluated with the help of various test cases.

Chapter 10

References and Bibliography

<https://spoonacular.com/food-api/docs#Search-Recipes-by-Ingredients>

<https://angular.io/docs>

<https://getbootstrap.com/docs/5.2/components/buttons/>

<https://nodejs.dev/learn/run-nodejs-scripts-from-the-command-line>

<https://ieeexplore.ieee.org/abstract/document/9438770>

<https://dl.acm.org/doi/abs/10.1145/3397271.3401244>

https://link.springer.com/chapter/10.1007/978-3-030-30645-8_55