

A Mini Project Report
On
RECIPEROLL



Submitted by

T. SWAPNA KIRANMAYI	21B91A05U3
U. PHANISWAR VARMA	21B91A05W1
B. KUSUMA SARANYA	22B95A0503
P. KAVYA	22B95A0525

DEPARTMENT OF COMPUTER SCIENCE ENGINEERING

S. R. K. R ENGINEERING COLLEGE (A)

(Affiliated to JNTUK, KAKINADA)

BHIMAVARAM-534204

[2023-2024]

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
S. R. K. R ENGINEERING COLLEGE BHIMAVARAM



CERTIFICATE

This is to certify that this is a bonafide work on “**RECIPEROLL**” for and has been submitted by **TADI SWAPNA KIRANMAYI** (21B91A05U3) , **UDDARAJU PHANISWAR VARMA** (21B91A05W1) , **BENDI KUSUMA SARANYA** (22B95A0503) & **PILLI KAVYA** (22B95A0525) as a **MEAN Stack Technologies and Data structures** laboratory report, in partial fulfilment of the requirements for the award of the degree of Bachelor of Technology in Computer Science and Engineering, during the academic year 2023-2024. The candidate worked right under my Supervision and Guidance

Lecturers In-Charge

Ch Ravi Swaroop
K D V Pavan Kumar
D V S R K Raju
Assistant professor,
Department of CSE,
S.R.K.R. Engineering College,
Bhimavaram.

Head of the Department

Dr. V. CHANDRA SEKHAR
Head of the Department,
Department of CSE,
S.R.K.R. Engineering College,
Bhimavaram.

ACKNOWLEDGEMENT

I take immense pleasure in thanking **Dr. K.V. Murali Krishnam Raju**, beloved principal of S.R.K.REngineering College, Bhimavaram, and **Dr. V. Chandra Sekhar**, esteemed **Head of the Department (CSE)** , for having permitted me to carry out this **MEAN Stack Technologies and Data Structures** project work.

I wish to express my deep sense of gratitude especially to my project Guide, **Sri. Ch Ravi Swaroop, Assistant Professor, Sri. K D V Pavan Kumar, Assistant Professor, Sri. D V S R K Raju, Assistant Professor**, for their guidance and useful suggestions, which helped me in completing the project work, on time.

Finally, yet importantly, I would like to express my heartfelt thanks to my beloved parents, faculty, my friends/classmates for their help, and my wishes for the successful completion of this project.

Tadi Swapna Kiranmayi
(21B91A05U3)

Uddaraju Phaniswar varma
(21B91A05W1)

Bendi Kusuma Saranya
(22B95A0503)

Pilli kavya
(22B95A0525)

TABLE OF CONTENTS

S. No	CONTENTS	Page. No
1	ABSTRACT	1
2	MODULES	2
3	SOFTWARE REQUIREMENTS	3
4	HARDWARE REQUIREMENTS	3
5	TECHNICAL STACK	4
6	PROJECT CODE	5-43
7	SCREENSHOTS	44-48

ABSTRACT

The recipe platform project is an online platform designed to provide users with a comprehensive database of recipes from various cuisines worldwide. The platform allows users to search for recipes based on specific criteria such as ingredients, cuisine type, dietary preferences, and meal type. Users can also browse through curated collections, trending recipes, and user-generated content. The platform features a user-friendly interface where users can view detailed recipe instructions, ingredient lists, cooking times. The backend of the platform consists of a database that stores all recipe information, including ingredients, instructions, images, and user data. The platform employs algorithms to organize and categorize recipes, making it easy for users to find relevant content quickly. The recipe platform project aims to provide a seamless and enjoyable user experience, catering to both novice cooks and seasoned chefs alike. It serves as a valuable resource for individuals looking to explore new culinary creations, enhance their cooking skills, and share their culinary experiences with others.

MODULES

- 1. User Registration and Profiles:** Allow users to create accounts, customize profiles, and manage their preferences.
- 2. Recipe Submission:** Enable users to submit their own recipes, including ingredients, instructions, and photos.
- 3. Recipe Discovery:** Provide various ways for users to discover new recipes, such as browsing categories, searching by ingredients, or exploring trending recipes.
- 4. Social Interaction:** Allow users to like, comment on, and share recipes with others. Incorporate social features like following other users, creating cooking circles, and private messaging.
- 5. Rating and Reviews:** Allow users to rate and review recipes they have tried, providing feedback for others in the community.

SOFTWARE REQUIREMENTS

- 1 Text Editor: Visual Studio Code
- 2 Front End: HTML, CSS , JAVASCRIPT, XML
- 3 Back End: MONGO DB,PHP
- 4 Internet Browser: Google Chrome/Mozilla Firefox/Internet Browser

HARDWARE REQUIREMENTS

- 1 Processor-Intel Core I5
- 2 RAM-8GB

TECHNICAL STACK

HTML

The **Hyper Text Markup Language** or **HTML** is the standard markup language for documents designed to be displayed in a web browser. HTML elements are the building blocks of HTML pages. Web browsers receive HTML documents from a web server or from local storage and render the documents into multimedia web pages. HTML describes the structure of a web page semantically and originally included cues for the appearance of the document. The file should be saved with extension .html.

CSS

Cascading Style Sheets (CSS) is a style sheet language used for describing the presentation of a document written in a markup language such as HTML or XML. CSS is designed to enable the separation of content and presentation, including layout, colors, and fonts. This separation can improve content accessibility; provide more flexibility and control in the specification of presentation characteristics; enable multiple web pages to share formatting by specifying the relevant CSS in a separate. The file should be saved with extension .css.

JAVA SCRIPT

JavaScript, often abbreviated as **JS**, is a programming language that is one of the core technologies of the World Wide Web, alongside HTML and CSS. As of 2022, 98% of websites use JavaScript on the client side for webpage behavior, often incorporating third-party libraries. JavaScript is a high-level, often just-in-time compiled language that conforms to the ECMA script standard. The JavaScript file should be saved with extension .js.

PHP

PHP is a popular programming language used for creating dynamic websites and web applications. It's known for its simplicity and flexibility, making it accessible for beginners and powerful for experienced developers. With PHP, developers can easily embed code within HTML, allowing for the creation of dynamic and interactive web pages. Additionally, PHP has a vast ecosystem of frameworks and libraries that streamline development tasks and enhance productivity. Its widespread adoption and active community support make it a reliable choice for building a wide range of web projects, from simple blogs to complex e-commerce platforms.

MONGO DB

MySQL is a popular database management system used to store and retrieve data for websites, applications, and more. It's like a digital filing cabinet, organizing information so it can be easily accessed and managed. With MySQL, users can create, read, update, and delete data, making it versatile for various needs. It's known for its reliability, scalability, and speed, making it a preferred choice for businesses of all sizes. Whether you're running a small blog or a large e-commerce platform, MySQL provides a solid foundation for storing and managing your data efficiently..

PROJECT CODE

Front end:

- index.ejs:

```
<div class="row flex-lg-row-reverse align-items-center g-5 py-4 mb-4">
<div class="col-12 col-lg-6">
  
</div>

<div class="col-12 col-lg-6">
  <h1 class="display-5 fw-bold mb-3">Huge selection of delicious recipe ideas</h1>
  <p class="lead">
    Explore our huge selection of delicious recipe ideas including; easy desserts, delicious vegan and
    vegetarian dinner ideas, gorgeous pasta recipes, quick bakes, family-friendly meals and gluten-free
    recipes.
  </p>

  <div class="d-grid gap-2 d-md-flex justify-content-md-start">
    <a href="/explore-latest" class="btn btn-primary btn-dark btn-lg px-4 me-md-2">Explore
    Latest</a>
    <a href="/explore-random" class="btn btn-outline-secondary btn-lg px-4 me-md-2">Show
    Random</a>
  </div>

</div>

</div>

<!-- Categories Start -->

<div class="row row-cols-2 row-cols-lg-6 g-2 g-lg-3 py-4">

  <% if(categories != "") { %>
    <% categories.forEach(function(category, index) { %>
      <a href="/categories/<% - category.name %>" class="col text-center category_link">
        <div class="category_img shadow">
          " loading="lazy">
        </div>
        <div class="pt-1"><% - category.name %></div>
      </a>
    <% }) %>
  <% } %>
```

```

<a href="/categories" class="col text-center category_link">
  <div class="category_img shadow">
    
  </div>
  <div class="pt-1">View All</div>
</a>

</div>
<!-- Categories End --&gt;

<!-- Latest Start --&gt;

&lt;section class="pb-4 pt-4"&gt;
  &lt;div class="d-flex mb-2 align-items-center"&gt;
    &lt;h2&gt;Latest Recipes&lt;/h2&gt;
    &lt;a href="/explore-latest" class="ms-auto"&gt;View More&lt;/a&gt;
  &lt;/div&gt;

&lt;div class="row row-cols-2 row-cols-lg-5 g-2 g-lg-3"&gt;

  &lt;% if(food != "") { %&gt;
    &lt;% food.latest.forEach(function(recipe, index){ %&gt;
      &lt;a href="/recipe/&lt;%= recipe.id %&gt;" class="col text-center category_link"&gt;
        &lt;div class="category_img category_img--large shadow"&gt;
          &lt;img src="/uploads/&lt;%= recipe.image %&gt;" alt="<%= recipe.name %>" loading="lazy">
        </div>
        <div class="pt-1"><%= recipe.name %></div>
      </a>
    <% }) %>
  <% } else { %>
    <p>No items found.</p>
  <% } %>
</div>

</section>
<!-- Latest End --&gt;

<!-- East Indian Recipes Start --&gt;
&lt;section class="pb-4 pt-4"&gt;
  &lt;div class="d-flex mb-2 align-items-center"&gt;
    &lt;h2&gt;East Indian Recipes&lt;/h2&gt;
    &lt;a href="/categories/Thai" class="ms-auto"&gt;View More&lt;/a&gt;
  &lt;/div&gt;

&lt;div class="row row-cols-2 row-cols-lg-5 g-2 g-lg-3"&gt;
</pre>

```

```

<% if(typeof food.thai !== 'undefined' && food.thai.length > 0) { %>
  <% food.thai.forEach(function(recipe, index){ %>
    <a href="/recipe/<%= recipe.id %>" class="col text-center category_link">
      <div class="category_img category_img--large shadow">
        " loading="lazy">
      </div>
      <div class="pt-1"><%= recipe.name %></div>
    </a>
  <% }) %>
<% } else { %>
  <p>No items found.</p>
<% } %>
</div>
</section>
<!-- East Indian Recipes End -->

<!-- South Indian Recipes Start -->
<section class="pb-4 pt-4">
  <div class="d-flex mb-2 align-items-center">
    <h2>South Indian Recipes</h2>
    <a href="/categories/American" class="ms-auto">View More</a>
  </div>

  <div class="row row-cols-2 row-cols-lg-5 g-2 g-lg-3">
    <% if(food.american != "") { %>
      <% food.american.forEach(function(recipe, index){ %>
        <a href="/recipe/<%= recipe.id %>" class="col text-center category_link">
          <div class="category_img category_img--large shadow">
            " loading="lazy">
          </div>
          <div class="pt-1"><%= recipe.name %></div>
        </a>
      <% }) %>
    <% } else { %>
      <p>No items found.</p>
    <% } %>
  </div>
</section>
<!-- South Indian Recipes End -->

<!-- North Indian Recipes -->
<section class="pb-4 pt-4">
  <div class="d-flex mb-2 align-items-center">
    <h2>North Indian Recipes</h2>
    <a href="/categories/Chinese" class="ms-auto">View More</a>

```

```

</div>

<div class="row row-cols-2 row-cols-lg-5 g-2 g-lg-3">
  <% if(food.chinese != "") { %>
    <% food.chinese.forEach(function(recipe, index){ %>
      <a href="/recipe/<%= recipe.id %>" class="col text-center category_link">
        <div class="category_img category_img--large shadow">
          " loading="lazy">
        </div>
        <div class="pt-1"><%= recipe.name %></div>
      </a>
    <% }) %>
  <% } else { %>
    <p>No items found.</p>
  <% } %>
  </div>
</section>
<!-- North Indian Recipes End -->

<!-- Submit - Start -->
<section class="px-4 py-5 my-5 text-center">
  
  <h1 class="display-5 fw-bold">Publish your recipe for FREE today</h1>
  <div class="col-lg-6 mx-auto">
    <p class="lead mb-4">Publish your Recipe in front of thousands of people for free.</p>
    <div class="d-grid gap-2 d-sm-flex justify-content-sm-center">
      <a href="/submit-recipe" class="btn btn-primary btn-dark btn-lg">Submit Recipe</a>
    </div>
  </div>
</section>

<!-- Submit - End -->

```

- about.ejs:

```

<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>About Us</title>
</head>
<body>
  <h1>About Us</h1>

```

<p>  Welcome to RecipeRoll, the ultimate destination for food enthusiasts to discover, and indulge in a world of flavours.

On RecipeRoll, you'll find a vibrant community of passionate cooks sharing their favourite recipes, from comforting classics to innovative creations.

</p>

<p> Whether you're seeking inspiration for your next meal or eager to share your culinary masterpieces with others, RecipeRoll offers a platform for connection and exploration.

RecipeRoll is more than just a recipe-sharing platform—it's a hub for culinary education, inspiration, and community.

</p>

<p> Join us on RecipeRoll and let's roll out the flavour together!      </p>

<p align="center">THANK YOU</p>

</body>

</html>

- categories.ejs:

```
<h1 class="pb-4">Explore Categories</h1>
<nav aria-label="breadcrumb">
  <ol class="breadcrumb">
    <li class="breadcrumb-item"><a href="/">Home</a></li>
    <li class="breadcrumb-item"><a href="/categories">Categories</a></li>
    <li class="breadcrumb-item active" aria-current="page">Explore Categories</li>
  </ol>
</nav>

<div class="row row-cols-2 row-cols-lg-5 g-2 g-lg-3 mb-4">
  <% if(typeof categories !== 'undefined' && categories.length > 0) { %>
    <% categories.forEach(function(category, index) { %>
      <a href="/categories/<%- category.name %>" class="col text-center category_link">
        <div class="category_img shadow">
          " loading="lazy">
        </div>
        <div class="pt-1"><%- category.name %></div>
      </a>
    <% }) %>
    <% } %>
  </div>

  <div class="row row-cols-2 row-cols-lg-5 g-2 g-lg-3">
    <% if(typeof categoryById !== 'undefined' && categoryById.length > 0) { %>
      <% categoryById.forEach(function(recipe, index){ %>
        <a href="/recipe/<%= recipe.id %>" class="col text-center category_link">
          <div class="category_img category_img--large shadow">
            " loading="lazy">
          </div>
```

```

<div class="pt-1"><%= recipe.name %></div>
</a>
<% }) %>
<% } else { %>
<p>No items found.</p>
<% } %>
</div>

```

- contact.ejs:

```

<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Contact Us - RecipeRoll</title>
  <style>
    body {
      font-family: Arial, sans-serif;
    }
    .container {
      max-width: 500px;
      margin: 0 auto;
      padding: 20px;
      border: 4px solid #0a0a0a;
      border-radius: 5px;
    }
    input[type="text"], input[type="email"], textarea {
      width: 100%;
      padding: 10px;
      margin-bottom: 10px;
      border: 3px solid #ccc;
      border-radius: 5px;
    }
    input[type="submit"] {
      background-color: #4CAF50;
      color: white;
      padding: 15px 20px;
      border: none;
      border-radius: 5px;
      cursor: pointer;
      font-size: 16px;
    }
    input[type="submit"]:hover {
      background-color: #45a049;
    }
  </style>
</head>
<body>
  <div class="container">
    <h1>Contact Us</h1>
    <form>
      <label>Name:</label>
      <input type="text" name="name" required="required"/>
      <label>Email:</label>
      <input type="email" name="email" required="required"/>
      <label>Message:</label>
      <textarea name="message" required="required">

```

```

        }
    </style>
</head>
<body>
<div class="container">
    <h2>Contact Us</h2>
    <p>Have a question, suggestion, or feedback? Feel free to get in touch with us!</p>
    <form action="submit_contact_form.php" method="post">
        <label for="name">Your Name</label>
        <input type="text" id="name" name="name" required>

        <label for="email">Your Email</label>
        <input type="email" id="email" name="email" required>

        <label for="message">Message</label>
        <textarea id="message" name="message" rows="4" required></textarea>

        <input type="submit" value="Submit">
    </form>
</div>
</body>
</html>

```

- explore-latest.ejs:

```

<h1 class="pb-4">Explore Latest</h1>

<div class="row row-cols-2 row-cols-lg-5 g-2 g-lg-3">
    <% if(recipe != "") { %>
        <% recipe.forEach(function(recipe, index){ %>
            <a href="/recipe/<%= recipe.id %>" class="col text-center category_link">
                <div class="category_img category_img--large shadow">
                    " loading="lazy">
                </div>
                <div class="pt-1"><%= recipe.name %></div>
            </a>
        <% }) %>
    <% } else { %>
        <p>No items found.</p>
    <% } %>
</div>

```

- explore-random.ejs:

```
<h1 class="pb-4">Explore Random</h1>
```

```

<div class="row row-cols-2 row-cols-lg-5 g-2 g-lg-3">
  <% if(recipe != "") { %>

    <a href="/recipe/<%= recipe.id %>" class="col text-center category_link">
      <div class="category_img category_img--large shadow">
        " loading="lazy">
      </div>
      <div class="pt-1"><%= recipe.name %></div>
    </a>

  <% } else { %>
    <p>No items found.</p>
  <% } %>
</div>

```

- recipe.ejs:

```

<% if(recipe != null ) { %>

  <nav aria-label="breadcrumb">
    <ol class="breadcrumb">
      <li class="breadcrumb-item"><a href="/">Home</a></li>
      <li class="breadcrumb-item active" aria-current="page"><%= recipe.name %></li>
    </ol>
  </nav>

  <div class="row">

    <div class="col-12 col-md-4">
      " loading="lazy">
    </div>

    <div class="col-12 col-md-8">
      <div class="row">
        <div class="col-12"><h1><%= recipe.name %></h1></div>
        <div class="col-12 mb-4"><i class="bi bi-tag"></i> <%= recipe.category %></div>
        <div class="col-12" style="white-space: pre-line;">
          <h4>Cooking Instructions</h4>
          <%= recipe.description %>
        </div>
      </div>
    </div>

    <div class="row pt-4">

```

```

<div class="col-12">
    <h4>Ingredients</h4>
    <ul class="list-group list-group-flush">
        <% recipe.ingredients.forEach(function(ingredients, index){ %>
            <li class="list-group-item"><%= ingredients %></li>
        <% }) %>
    </ul>
</div>
</div>
</div>
</div>
<% } else { %>
    <p>No item found.</p>
<% } %>

```

- search.ejs:

```

<h1 class="pb-4">Search Results</h1>

<div class="row row-cols-2 row-cols-lg-5 g-2 g-lg-3">
<% if(recipe != '') { %>
    <% recipe.forEach(function(recipe, index){ %>
        <a href="/recipe/<%= recipe.id %>" class="col text-center category_link">
            <div class="category_img category_img--large shadow">
                " loading="lazy">
            </div>
            <div class="pt-1"><%= recipe.name %></div>
        </a>
    <% }) %>
<% } else { %>
    <p>No items found.</p>
<% } %>
</div>

```

- submit-recipe.ejs:

```

<div class="px-4 py-5 my-5 text-center">
    <h1 class="display-5 fw-bold">Submit Your Recipe</h1>
    <div class="col-lg-6 mx-auto">
        <p class="lead">Share your amazing recipes with thousands of web developers accross the world.
            Fill our form to get started.</p>
    </div>
</div>

<div class="row justify-content-center">
<% if(infoSubmitObj != '') { %>

```

```

<div class="col-8 alert alert-success" role="alert">
    <%= infoSubmitObj %>
</div>
<% } %>

<% if(infoErrorsObj != "") { %>
<div class="col-8 alert alert-danger" role="alert">
    <%= infoErrorsObj[0].message %>
</div>
<% } %>

<div class="col-8">

<form action="/submit-recipe" enctype="multipart/form-data" method="POST">

<div class="row g-3">

    <div class="col-12">
        <label for="email" class="form-label">Email</label>
        <input type="email" name="email" id="email" class="form-control">
    </div>

    <div class="col-12">
        <label for="name" class="form-label">Recipe Name</label>
        <input type="text" name="name" id="name" class="form-control" required>
    </div>

    <div class="col-12">
        <label for="description" class="form-label">Description</label>
        <textarea name="description" id="description" class="form-control" cols="30" rows="4"></textarea>
    </div>

    <div class="col-12">
        <label for="name" class="form-label">Ingredients</label><br>
        <small>Example: Ice</small>
        <div class="ingredientList">
            <div class="ingredieintDiv mb-1">
                <input type="text" name="ingredients" class="form-control">
            </div>
        </div>
    </div>

    <div class="col-12">
        <button type="button" class="btn btn-outline-primary" id="addIngredientsBtn">+
        Ingredient</button>
    </div>

```

```

<div class="col-12">
  <label for="category">Select Category</label>
  <select class="form-select form-control" name="category" aria-label="Category">
    <option selected>Select Category</option>
    <option value="Thai">East</option>
    <option value="American">South</option>
    <option value="Chinese">North</option>
    <option value="Mexican">West</option>
    <option value="Indian">Fast Food</option>
  </select>
</div>

<div class="col-12">
  <label for="image">Product Image</label>
  <input type="file" class="form-control" name="image" accept="image/*">
</div>

<div class="col-12">
  <button type="submit" class="btn btn-primary">Submit Recipe</button>
</div>

</div>
</div>

```

- app.js:

```

const express = require('express');
const expressLayouts = require('express-ejs-layouts');
const fileUpload = require('express-fileupload');
const session = require('express-session');
const cookieParser = require('cookie-parser');
const flash = require('connect-flash');

const app = express();
const port = process.env.PORT || 3000;

require('dotenv').config();

app.use(express.urlencoded( { extended: true } ));
app.use(express.static('public'));
app.use(expressLayouts);

app.use(cookieParser('RecipeRollSecure'));
app.use(session({

```

```

secret: 'RecipeRollSecretSession',
saveUninitialized: true,
resave: true
}});
app.use(flash());
app.use(fileUpload());

app.set('layout', './layouts/main');
app.set('view engine', 'ejs');

// Import the controller directly
const recipeController = require('./server/controllers/recipeController');

// Define the contact route
app.get('/contact', recipeController.contactPage);
app.get('/about', recipeController.aboutPage);

// Import routes from recipeRoutes.js
const routes = require('./server/routes/recipeRoutes.js');
app.use('/', routes);

app.listen(port, () => console.log(Listening to port ${port})));

```

- package.json:

```
{
  "name": "reciperoll",
  "version": "1.0.0",
  "description": "",
  "main": "index.js",
  "scripts": {
    "test": "echo \\\"Error: no test specified\\\" && exit 1",
    "start": "nodemon app.js"
  },
  "keywords": [],
  "author": "",
  "license": "ISC",
  "dependencies": {
    "connect-flash": "^0.1.1",
    "cookie-parser": "^1.4.5",
    "dotenv": "^10.0.0",
    "ejs": "^3.1.6",
    "express": "^4.17.1",
    "express-ejs-layouts": "^2.5.1",
    "express-fileupload": "^1.2.1",
    "express-session": "^1.17.2",
  }
}
```

```
    "mongodb": "^4.1.2",
    "mongoose": "^6.4.6"
  },
  "devDependencies": {
    "nodemon": "^2.0.13"
  }
}
```

- style.js:

```
:root {
  --bs-font-sans-serif: 'Rubik', sans-serif;
}

.container-xxl { max-width: 1530px; }

body {
  background-image: url("/img/i-like-food.svg");
  background-color: rgb(243,243,243);
}

a { color: var(--bs-dark); }

/* Category Card */
.category_link {
  text-align: center;
  display: block;
  text-decoration: none;
  transition: 250ms all;
}

.category_link:hover {
  transform: scale(1.1);
}

.category_img {
  display: flex;
  justify-content: center;
  align-items: center;
  height: 100px;
  margin-bottom: 10px;
  box-shadow: 0px 3px 6px rgba(0,0,0,.16);
  overflow: hidden;
  border-radius: 8px;
}

.category_img--large {
```

```
height: 330px;  
}
```

```
.category_img img {  
    width: 100%;  
    object-fit: cover;  
}
```

- recipecontroller.js:

```
require('../models/database');  
const Category = require('../models/Category');  
const Recipe = require('../models/Recipe');  
  
/**  
 * GET /  
 * Homepage  
 */  
exports.homepage = async(req, res) => {  
    try {  
        const limitNumber = 5;  
        const categories = await Category.find({ }).limit(limitNumber);  
        const latest = await Recipe.find({ }).sort({ _id: -1 }).limit(limitNumber);  
        const thai = await Recipe.find({ 'category': 'Thai' }).limit(limitNumber);  
        const american = await Recipe.find({ 'category': 'American' }).limit(limitNumber);  
        const chinese = await Recipe.find({ 'category': 'Chinese' }).limit(limitNumber);  
  
        const food = { latest, thai, american, chinese };  
  
        res.render('index', { title: 'RecipeRoll - Home', categories, food } );  
    } catch (error) {  
        res.status(500).send({ message: error.message || "Error Occured" });  
    }  
}  
  
/**  
 * GET /categories  
 * Categories  
 */  
exports.exploreCategories = async(req, res) => {  
    try {  
        const limitNumber = 20;  
        const categories = await Category.find({ }).limit(limitNumber);  
        res.render('categories', { title: 'RecipeRoll - Categories', categories } );  
    } catch (error) {  
        res.status(500).send({ message: error.message || "Error Occured" });  
    }  
}
```

```

/**
 * GET /categories/:id
 * Categories By Id
 */
exports.exploreCategoriesById = async(req, res) => {
  try {
    let categoryId = req.params.id;
    const limitNumber = 20;
    const categoryById = await Recipe.find({ 'category': categoryId }).limit(limitNumber);
    res.render('categories', { title: 'RecipeRoll - Categories', categoryById } );
  } catch (error) {
    res.status(500).send({message: error.message || "Error Occured" });
  }
}

/**
 * GET /recipe/:id
 * Recipe
 */
exports.exploreRecipe = async(req, res) => {
  try {
    let recipeId = req.params.id;
    const recipe = await Recipe.findById(recipeId);
    res.render('recipe', { title: 'RecipeRoll - Recipe', recipe } );
  } catch (error) {
    res.status(500).send({message: error.message || "Error Occured" });
  }
}

/**
 * POST /search
 * Search
 */
exports.searchRecipe = async(req, res) => {
  try {
    let searchTerm = req.body.searchTerm;
    let recipe = await Recipe.find( { $text: { $search: searchTerm, $diacriticSensitive: true } } );
    res.render('search', { title: 'RecipeRoll - Search', recipe } );
  } catch (error) {
    res.status(500).send({message: error.message || "Error Occured" });
  }
}

/**

```

```

/* GET /explore-latest
 * Explplore Latest
 */
exports.exploreLatest = async(req, res) => {
  try {
    const limitNumber = 20;
    const recipe = await Recipe.find({ }).sort({ _id: -1 }).limit(limitNumber);
    res.render('explore-latest', { title: 'RecipeRoll - Explore Latest', recipe } );
  } catch (error) {
    res.satus(500).send({message: error.message || "Error Occured" });
  }
}

/***
 * GET /explore-random
 * Explore Random as JSON
 */
exports.exploreRandom = async(req, res) => {
  try {
    let count = await Recipe.find().countDocuments();
    let random = Math.floor(Math.random() * count);
    let recipe = await Recipe.findOne().skip(random).exec();
    res.render('explore-random', { title: 'RecipeRoll - Explore Latest', recipe } );
  } catch (error) {
    res.satus(500).send({message: error.message || "Error Occured" });
  }
}

/***
 * GET /submit-recipe
 * Submit Recipe
 */
exports.submitRecipe = async(req, res) => {
  const infoErrorsObj = req.flash('infoErrors');
  const infoSubmitObj = req.flash('infoSubmit');
  res.render('submit-recipe', { title: 'RecipeRoll - Submit Recipe', infoErrorsObj, infoSubmitObj } );
}

/***
 * POST /submit-recipe
 * Submit Recipe
 */
exports.submitRecipeOnPost = async(req, res) => {
  try {

```

```

let imageUploadFile;
let uploadPath;
let newImageName;

if(!req.files || Object.keys(req.files).length === 0){
  console.log('No Files where uploaded.');
} else {

  imageUploadFile = req.files.image;
  newImageName = Date.now() + imageUploadFile.name;

  uploadPath = require('path').resolve('./') + '/public/uploads/' + newImageName;

  imageUploadFile.mv(uploadPath, function(err){
    if(err) return res.status(500).send(err);
  })
}

const newRecipe = new Recipe({
  name: req.body.name,
  description: req.body.description,
  email: req.body.email,
  ingredients: req.body.ingredients,
  category: req.body.category,
  image: newImageName
});

await newRecipe.save();

req.flash('infoSubmit', 'Recipe has been added.')
res.redirect('/submit-recipe');
} catch (error) {
  // res.json(error);
  req.flash('infoErrors', error);
  res.redirect('/submit-recipe');
}
}

/***
 * GET /about
 * Contact Page
 */
exports.aboutPage = async (req, res) => {
try {
  res.render('about', { title: 'RecipeRoll - About Us' });
} catch (error) {

```

```

        res.status(500).send({ message: error.message || "Error Occurred" });
    }
}

/***
 * GET /contact
 * Contact Page
 */
exports.contactPage = async (req, res) => {
    try {
        res.render('contact', { title: 'RecipeRoll - Contact Us' });
    } catch (error) {
        res.status(500).send({ message: error.message || "Error Occurred" });
    }
}

```

```

// Delete Recipe
// async function deleteRecipe(){
//   try {
//     await Recipe.deleteOne({ name: 'New Recipe From Form' });
//   } catch (error) {
//     console.log(error);
//   }
// }
// deleteRecipe();

```

```

// Update Recipe
// async function updateRecipe(){
//   try {
//     const res = await Recipe.updateOne({ name: 'New Recipe' }, { name: 'New Recipe Updated' });
//     res.n; // Number of documents matched
//     res.nModified; // Number of documents modified
//   } catch (error) {
//     console.log(error);
//   }
// }
// updateRecipe();

```

```

/***
 * Dummy Data Example
 */
// async function insertDymmyCategoryData(){
//   try {

```

```

// await Category.insertMany([
//   {
//     "name": "Thai",
//     "image": "thai-food.jpg"
//   },
//   {
//     "name": "American",
//     "image": "american-food.jpg"
//   },
//   {
//     "name": "Chinese",
//     "image": "chinese-food.jpg"
//   },
//   {
//     "name": "Mexican",
//     "image": "mexican-food.jpg"
//   },
//   {
//     "name": "Indian",
//     "image": "indian-food.jpg"
//   },
//   {
//     "name": "Spanish",
//     "image": "spanish-food.jpg"
//   }
// ]);

// } catch (error) {
//   console.log('err', + error)
// }
// }

// insertDymmyCategoryData();

// async function insertDymmyRecipeData(){
// try {
//   await Recipe.insertMany([
//     {
//       "name": "Recipe Name Goes Here",
//       "description": Recipe Description Goes Here,
//       "email": "recipeemail@raddy.co.uk",
//       "ingredients": [
//         "1 level teaspoon baking powder",
//         "1 level teaspoon cayenne pepper",
//         "1 level teaspoon hot smoked paprika",
//       ],
//       "category": "American",
//       "image": "southern-friend-chicken.jpg"
//     }
//   ]);
// } catch (error) {
//   console.log('err', + error)
// }
// }

```

```

//     },
//     {
//       "name": "Recipe Name Goes Here",
//       "description": Recipe Description Goes Here,
//       "email": "recipeemail@raddy.co.uk",
//       "ingredients": [
//         "1 level teaspoon baking powder",
//         "1 level teaspoon cayenne pepper",
//         "1 level teaspoon hot smoked paprika",
//       ],
//       "category": "American",
//       "image": "southern-friend-chicken.jpg"
//     },
//   ]);
// } catch (error) {
//   console.log('err', + error)
// }
// }

// insertDymmyRecipeData();

```

- category.js:

```

const mongoose = require('mongoose');

const categorySchema = new mongoose.Schema({
  name: {
    type: String,
    required: 'This field is required.'
  },
  image: {
    type: String,
    required: 'This field is required.'
  },
});

module.exports = mongoose.model('Category', categorySchema);

```

- recipe.js:

```

const mongoose = require('mongoose');

const recipeSchema = new mongoose.Schema({
  name: {
    type: String,
    required: 'This field is required.'
  },
  description: {

```

```

    type: String,
    required: 'This field is required.'
},
email: {
    type: String,
    required: 'This field is required.'
},
ingredients: {
    type: Array,
    required: 'This field is required.'
},
category: {
    type: String,
    enum: ['Thai', 'American', 'Chinese', 'Mexican', 'Indian'],
    required: 'This field is required.'
},
image: {
    type: String,
    required: 'This field is required.'
},
});
};

recipeSchema.index({ name: 'text', description: 'text' });
// WildCard Indexing
//recipeSchema.index({ "$**" : 'text' });

module.exports = mongoose.model('Recipe', recipeSchema);

```

- database.js:

```

const mongoose = require('mongoose');
mongoose.connect(process.env.MONGODB_URI, { useNewUrlParser: true, useUnifiedTopology: true
});

const db = mongoose.connection;
db.on('error', console.error.bind(console, 'connection error:'));
db.once('open', function(){
    console.log('Connected')
});

// Models
require('./Category');
require('./Recipe');

```

- reciperoute.js:

```

const express = require('express');
const router = express.Router();
const recipeController = require('../controllers/recipeController');

```

```

/**
 * App Routes
 */
router.get('/', recipeController.homepage);
router.get('/recipe/:id', recipeController.exploreRecipe );

router.get('/categories', recipeController.exploreCategories);
router.get('/categories/:id', recipeController.exploreCategoriesById);
router.post('/search', recipeController.searchRecipe);
router.get('/explore-latest', recipeController.exploreLatest);
router.get('/explore-random', recipeController.exploreRandom);
router.get('/submit-recipe', recipeController.submitRecipe);
router.post('/submit-recipe', recipeController.submitRecipeOnPost);
router.post('/contact', recipeController.contactPage);
router.post('/about', recipeController.aboutPage);

module.exports = router;

```

- main.js:

```

<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<meta http-equiv="X-UA-Compatible" content="IE=edge">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title><%= typeof title != 'undefined' ? title: 'RecipeRoll' %></title>
<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.1/dist/css/bootstrap.min.css" rel="stylesheet"
      integrity="sha384-
F3w7mX95PdgyTmZZMECAngseQB83DfGTowi0iMjiWaeVhAn4FJkqJBByhZMI3AhiU"
      crossorigin="anonymous">
<link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap-icons@1.5.0/font/bootstrap-
icons.css">
<link rel="preconnect" href="https://fonts.googleapis.com">
<link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>
<link href="https://fonts.googleapis.com/css2?family=Rubik:wght@300;500&display=swap"
      rel="stylesheet">
<link rel="stylesheet" href="/css/styles.css">
</head>
<body>

<div class="container-xxl px-md-5 bg-white shadow-lg">

<header class="d-flex flex-wrap align-items-center justify-content-center justify-content-md-
between py-3 mb-4 border-bottom">
<a href="/" class="d-flex align-items-center col-md-3 mb-2 mb-md-0 text-dark text-decoration-

```

```

    none">
    
</a>

<ul class="nav col-12 col-md-auto mb-2 justify-content-center mb-md-0">
<li><a href="/" class="nav-link px-2 link-secondary">Home</a></li>
<li><a href="/about" class="nav-link px-2 link-secondary">About</a></li>
<li><a href="/submit-recipe" class="nav-link px-2 link-secondary">Submit</a></li>
<li><a href="/contact" class="nav-link px-2 link-secondary">Contact</a></li>
</ul>

<div class="col-md-3 text-end">
    <form method="POST" action="/search">
        <input type="search" name="searchTerm" class="form-control" placeholder="Search..." aria-label="Search">
    </form>
</div>
</header>

<main>
    <% - body -%>
</main>

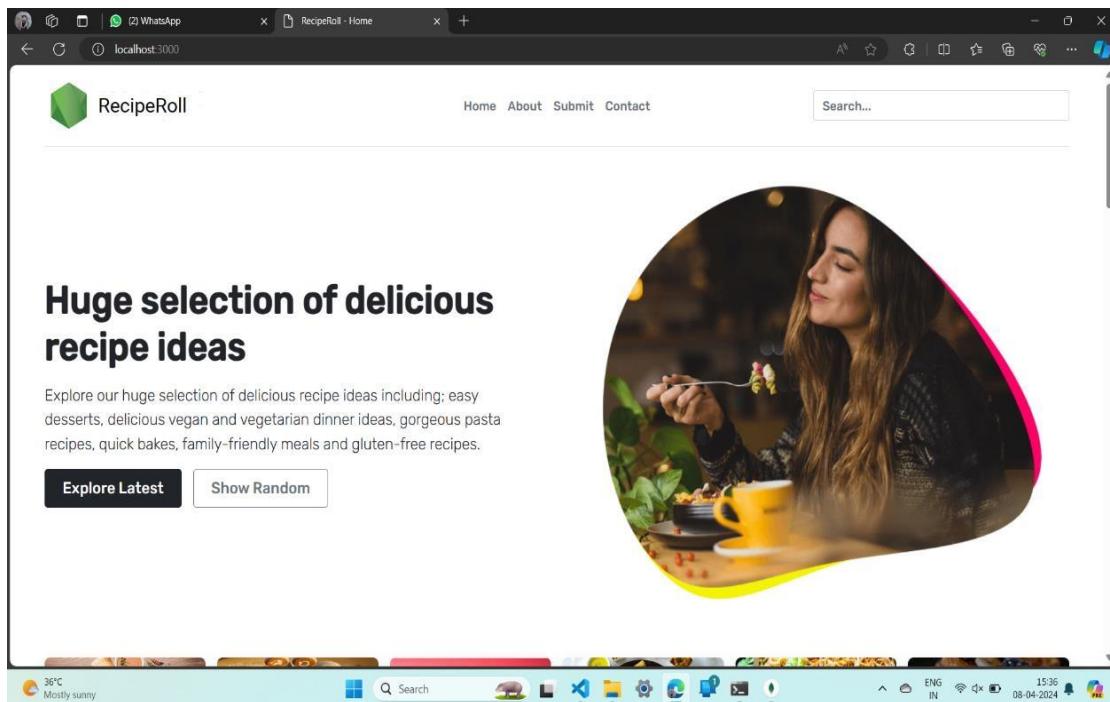
</div>

<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.1.1/dist/js/bootstrap.bundle.min.js"
    integrity="sha384-/bQdsTh/da6pkI1MST/rWKFNjaCP5gBSY4sEBT38Q/9RBh9AH40zEOg7Hlq2THRZ"
    crossorigin="anonymous"></script>
<script src="/js/script.js"></script>
</body>
</html>

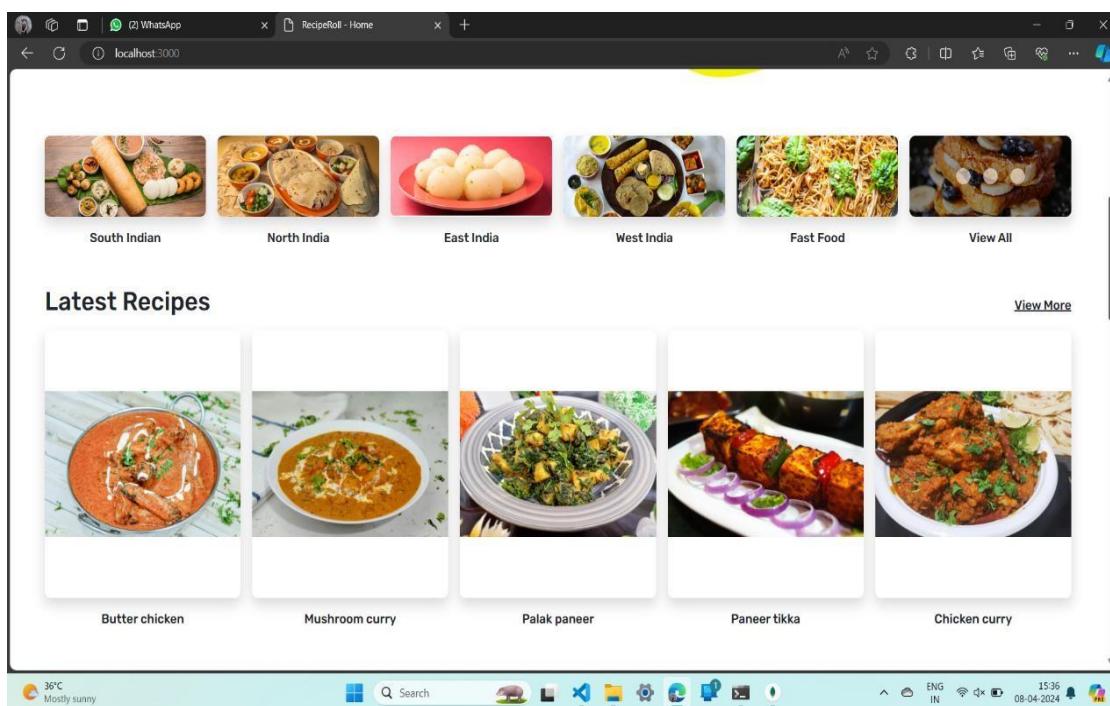
```

SCREENSHOTS

Main page:



Latest recipes:

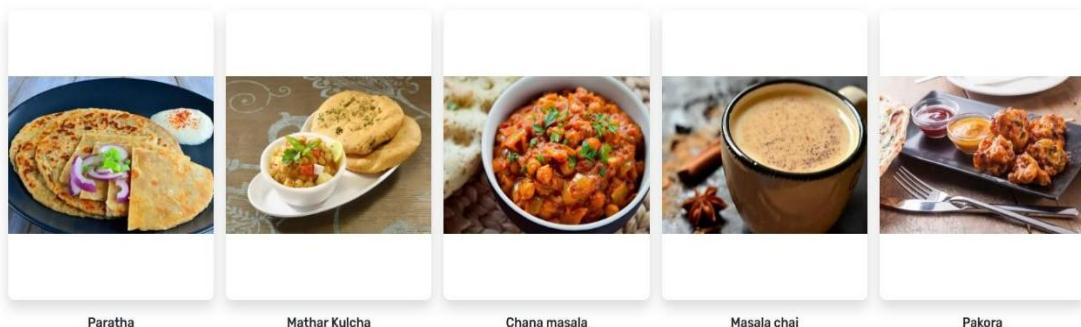


Explore categories:

The screenshot shows a web browser window for 'RecipeRoll - Categories' at 'localhost:3000/categories'. The page features a navigation bar with 'Home', 'About', 'Submit', and 'Contact' links, and a search bar. The main content area is titled 'Explore Categories' and includes a breadcrumb trail: 'Home / Categories / Explore Categories'. Below this are five category sections with images and labels: 'South Indian' (a variety of Indian breads), 'North India' (tacos and tortillas), 'East India' (a plate of white dumplings), 'West India' (a plate of Indian flatbreads), and 'Fast Food' (noodles). A separate section for 'Deserts' shows a chocolate cake. A message 'No items found.' is displayed below the categories. The bottom of the screen shows a Windows taskbar with icons for weather (36°C, mostly sunny), search, file explorer, and system settings, along with the date and time (08-04-2024, 15:36).

East Indian Recipes

[View More](#)



South Indian Recipes

[View More](#)



North Indian Recipes

[View More](#)



Random recipes:

Home / Categories / Explore Categories

Coconut rice Idli Pongal Appam Chicken curry

Submit recipe:

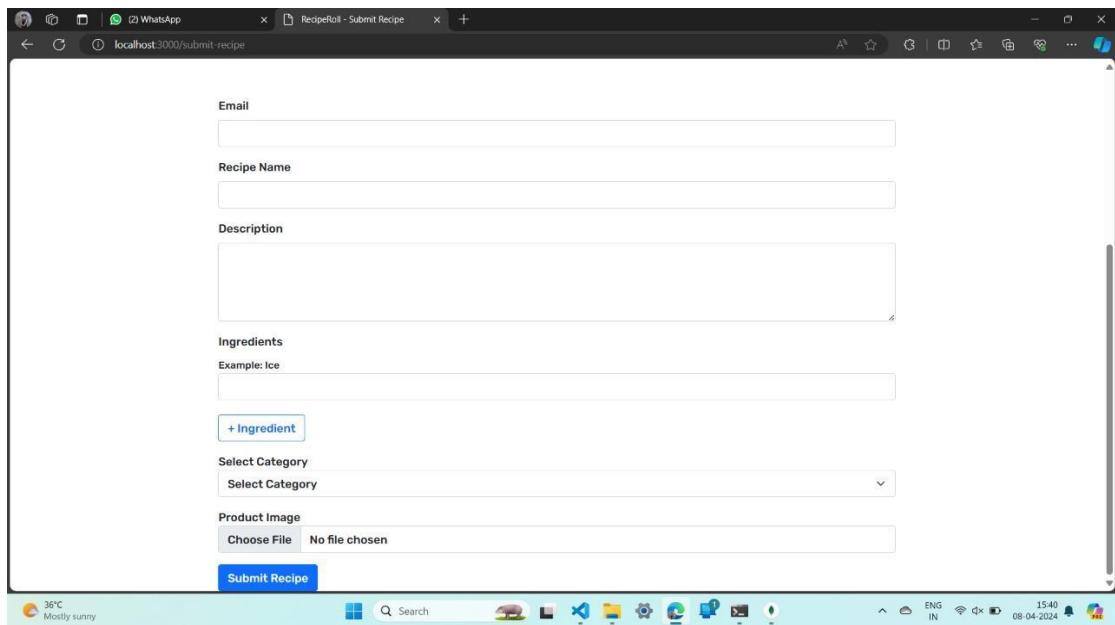
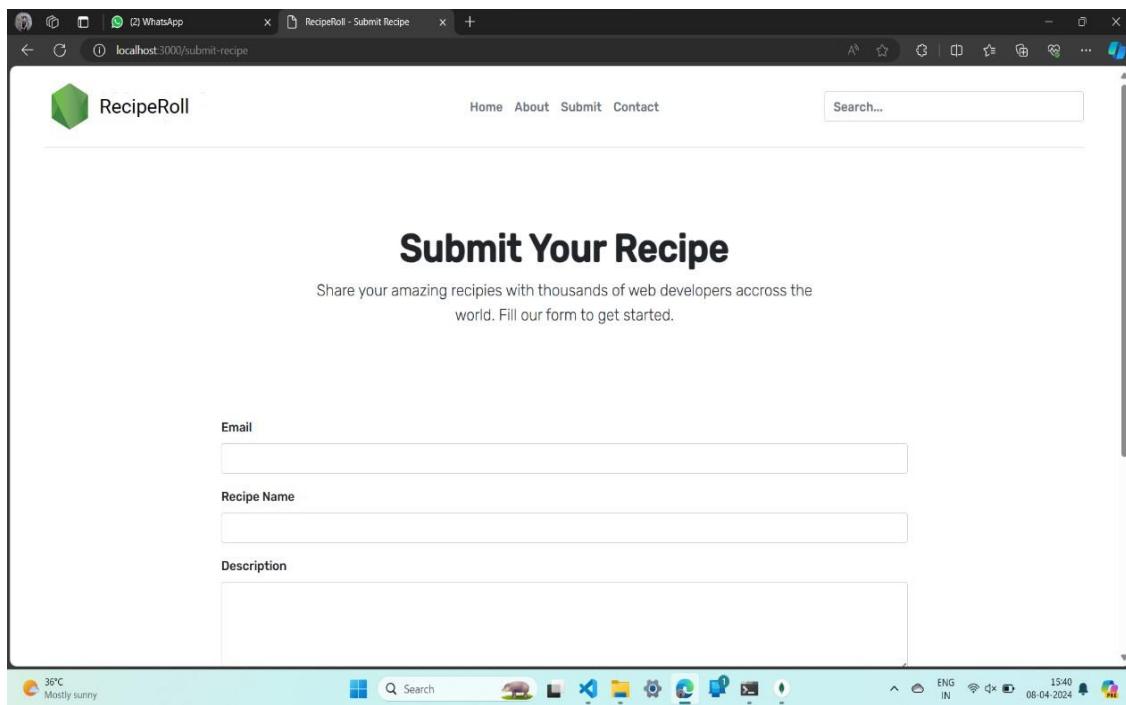


Publish your recipe for FREE today

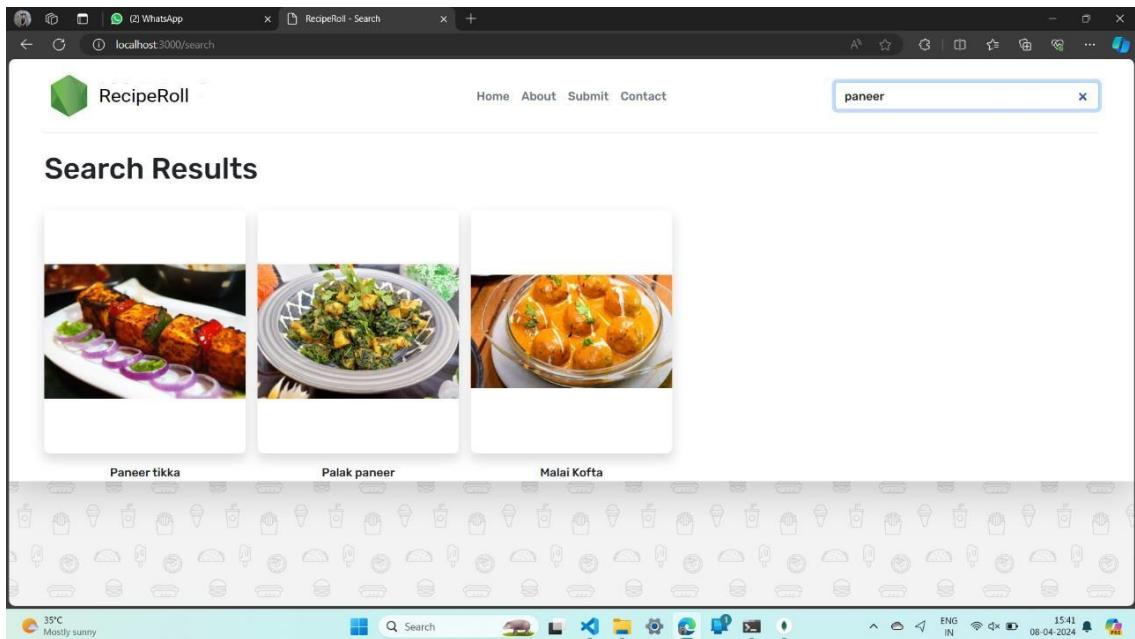
Publish your Recipe in front of thousands of people for free.

[Submit Recipe](#)

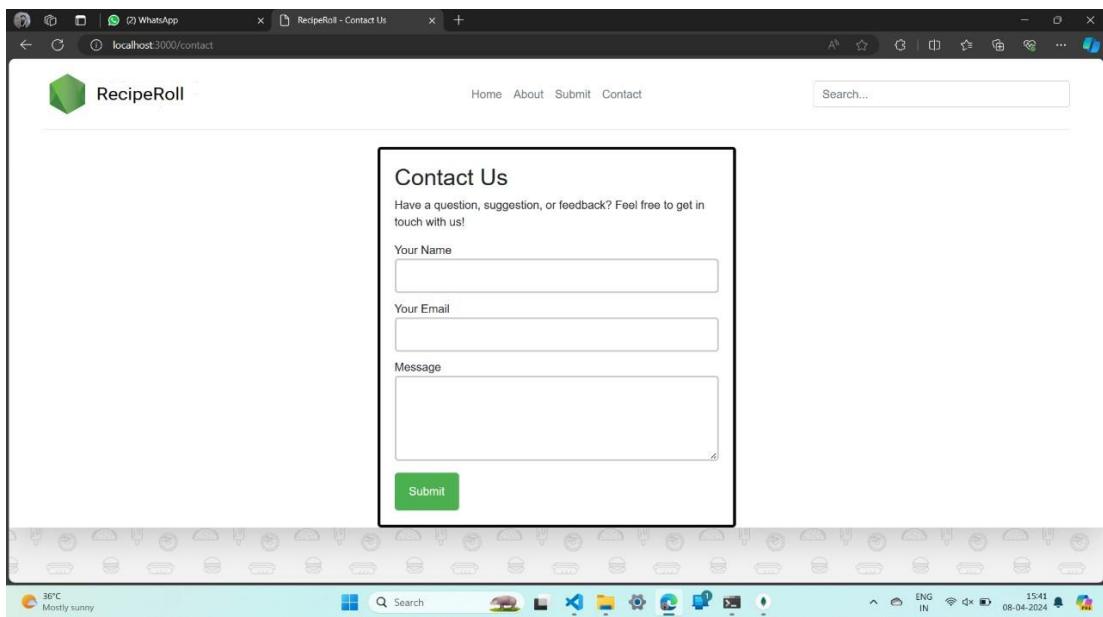
Recipe sharing page:



Search recipes:



Contact us page:



About us page:

