VISVESVARAYA TECHNOLOGICAL UNIVERSITY

Jnana Sangama, Belagavi-590018

AngularJS and NodeJs Mini Project Report on

"Recipe App"

Submitted in Partial fulfillment of the Requirements for the V Semester of the Degree of

Bachelor of Engineering in

Information Science & Engineering

By

ANKITH CHENGAPPA B (1CR21IS024)

GOUTHAM GS NAIR (1CR21IS056)

HARISH M (1CR21IS057)

Under the Guidance of,

Mrs. K KOMALA DEVI, Assistant Professor, Dept. of ISE

DEPARTMENT OF INFORMATION SCIENCE AND ENGINEERING

CMR INSTITUTE OF TECHNOLOGY

Affiliated to VTU, Approved by AICTE, Accredited by NBA and NAAC with "A++" Grade

ITPL MAIN ROAD, BROOKFIELD, BENGALURU-560037, KARNATAKA, INDIA

CERTIFICATE

This is to certify that the AngularJS and NodeJS Project work entitled **“Recipe App”** has been carried out by Ankith Chengappa B, Goutham GS Nair, Harish M bonafide studentsof CMR Institute of Technology, Bengaluru in partial fulfillment for the award of the Degree of **Bachelor of Engineering in Information Science and Engineering** of the Visvesvaraya Technological University, Belagavi during the year **2023-2024**. It is certified that all corrections/suggestions indicated for the Internal Assessment have been incorporated in the report deposited in the departmental library. This Project report has been approved as it satisfies the academic requirements in respect of project work prescribed for the said Degree.

Signature of Guide	**Signature of HOD**
Mrs K Komala Devi	**Dr Jagadishwari V**
Assistant Professor	**Professor & HoD**
Dept. of ISE, CMRIT	**Dept. of ISE, CMRIT**

External Viva

Name of the Examiners Signature with date

1.

2.

DECLARATION

We, the students of V semester from Department of Information Science and Engineering, CMR Institute of Technology, Bangalore declare that the project work entitled "**Recipe App”** has been successfully completed under the guidance of Mrs. K Komala Devi, Assistant Professor, Dept. of Information Science and Engineering, CMR Institute of technology, Bengaluru. This project work is submitted in partial fulfillment of the requirements for the award of the Degree of Bachelor of Engineering in Information Science and Engineering during the academic year 2023-2024. The matter embodied in the project report has not been submitted previously by anybody for the award of any degree or diploma to any university.

Place: Bangalore

Date: 0503/2024

Team members:

ANKITH CHENGAPPA B(1CR21IS024)	
GOUTHAM GS NAIR(1CR21IS056)	
HARISH M(1CR21IS057)	

ABSTRACT

The Web Based Cook Book is a single page web-based recipe application developed using Angular, aimed at facilitating users in creating, updating, and managing recipes seamlessly, coupled with a dynamic shopping list module enabling users to manage their grocery needs efficiently. Users can add ingredients directly of their own to the shopping list further simplifying the meal preparation process. Additionally, the shopping list supports functionalities such as item deletion and bulk importing of ingredients from selected recipes. Leveraging Angular' s robust framework, the application offers an intuitive user interface which allows users to switch from the shopping list to the recipe list and vice-versa and also ensures a smooth and responsive experience. Overall, the Web Based Cook Book aims to enhance the culinary experience for users, empowering them to create and organize recipes effortlessly while streamlining the grocery shopping process.

ACKNOWLEDGEMENT

I take this opportunity to express my sincere gratitude and respect to **CMR Institute of Technology, Bengaluru** for providing me a platform to pursue my studies and carry out the Database Management System Project.

It gives me an immense pleasure to express my deep sense of gratitude to **Dr. Sanjay Jain,** Principal, CMRIT, Bengaluru, for his constant encouragement.

I would like to extend my sincere gratitude to **Dr. Jagadishwari V,** HOD, Department of Information Science and Engineering, CMRIT, Bengaluru, who has been a constant support and encouragement throughout the course of this project.

I would like to thank my guide **Mrs. K Komala Devi, Assistant Professor,** Department of Information Science and Engineering, for the valuable guidance throughout the tenure of the project work.

I would also like to thank all the faculty members of Department of Information Science and Engineering who directly or indirectly encouraged me.

Finally, I thank my parents and friends for all the moral support they have given me during the completion of this work.

TABLE OF CONTENTS

Contents	Page No.
Certificate	ii
Declaration	iii
Abstract	iv
Acknowledgement	v
Table of contents	vi
List of Figures	vii
1. Introduction	1
2. System Requirements 2.1 Hardware Requirements 2.2 Software Requirements	4
3. Implementation	6
4. Output	19
5. Conclusion and Future Scope	24
6. References	26

LIST OF FIGURES

Page No.

Fig 1.1 DataFlow 2

Fig 4.2-4.5 Code Snippets 6-10

Fig 5.1-5.3 Outputs 11-13

CHAPTER 1

INTRODUCTION

Recipe App is a very useful app for people who love to cook and try out new recipes. Recipe App application provides user flexibility to search, share, save recipes from cloud with an additional capability to maintain personal cookbook for creating new recipe, deleting recipe that are no longer required. Our application “Recipe App” makes finding recipes easy. Recipes are from everyday cooks and chefs that have perfected the recipe over time. Firebase is used for Storing recipes from cloud. Recipes range from the decadent to the simple for whatever mood user are in. With recipes being added daily in database there will always be something new for user to try. Recipes are from well-known publishers All Recipes, 101 Cookbooks, Closet Cooking and many more. Recipe App Application provides capability to user to search top rated, trending recipes. Recipes are displayed with recipe image, title, ingredients list and cooking directions. Category based and ingredients based search is also provided to user.. The user is given choice to create personal cookbook, where user can create recipe, view recipe and delete recipe. The interface is clean and simple The user can search recipes, view added favorite recipe list and access personal cookbook all from home screen..

Recipe application save recipes added by user in database providing capability to view saved recipes when required. When user clicks personal cookbook image button on home page, user gets image button for viewing recipe. If recipe image is not provided by user during creation of recipe default image gets displayed with message no image available. User can select recipe from list by clicking on name. Upon click recipe title, time required for cooking, ingredients needed, cooking directions and recipe image are displayed.

User has delete option in personal cookbook screen. User can click image button with cross sign for performing deletion. Upon click, user saved recipe list gets displayed with image and title. User can select recipe from list, on selection alert dialog box is displayed with message deleting recipe. User can select yes button for deletion, and no to cancel delete operation. When user clicks yes, recipe gets deleted and page is refreshed. Upon clicking no, user is navigated to recipe list screen.

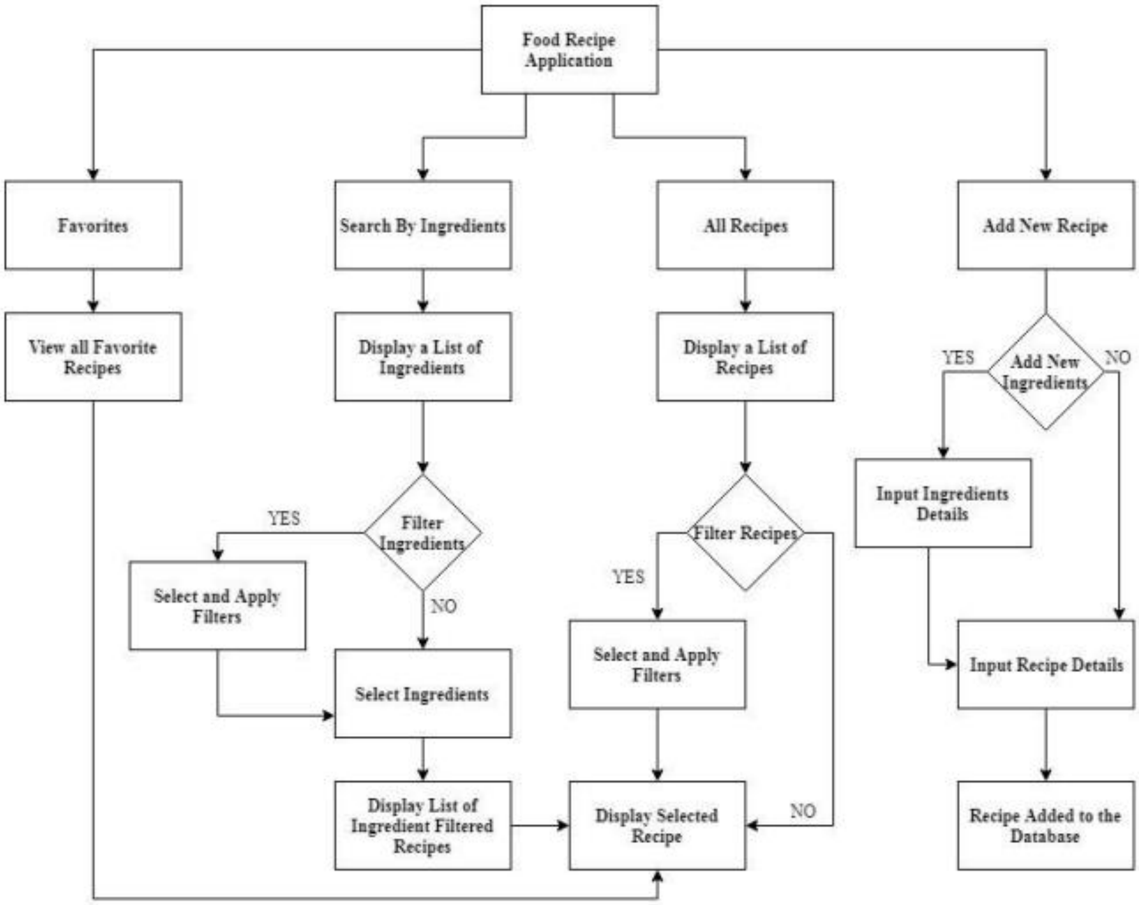


Fig 1.1 DataFlow Diagram

1.1 Aims and Objectives

The main aim of our application is to provide recipes to the consumers based on the ingredients already available with them, unlike other recipe providing applications where the ingredients available with the consumer is not taken into consideration. The objectives of our project are as follows:

- To help the user decide a recipe to cook from the ingredients available with him/her.
- To guide the user to the recipe based on the user’s choices and needs.
- To help save the user money and time by tediously referencing cook books and buying ingredients he/she does not need

1.2 Problem Statement

We all have those times when we don’t know what we could make for ourselves to eat from what we have available with us. Even if we do we, may not know about a new recipe that can be made from the same ingredients or an old recipe that can be made in a different way. Today there are innumerable applications that provide consumers with recipes ranging from quick to healthy and from beginner to expert; all intending to save time. But, none of these applications take into account whether he recipes ingredients are available with the consumer at the point of time or not. They fail to provide recipes containing only the ingredients that are available, thus proving to be inefficient and wasting time rather than saving it. These contemporary applications also do not evaluate and learn from the user’s choices thus further increasing the user’s task of repeating already fed information again. We plan on using a content-based recommendation system that will learn from user’s inputs and provide the user with Reccomendations.

CHAPTER 2

SYSTEM REQUIREMENTS

For an e-commerce application, the system requirements can vary based on the scale and complexity of the project. Here's a general outline of the hardware and software requirements you might consider:

1. **Hardware Requirements:**
   - **Web Server**: A server capable of hosting the application, handling HTTP requests, and serving web pages.
   - **Database Server**: A server capable of hosting the database management system (DBMS) to store product information, user data, and order details.
   - **Storage**: Sufficient storage space for storing images, product descriptions, and other media files.
   - **Backup System**: A mechanism for regular backups to prevent data loss.

2. **Software Requirements:**
   - **Operating System**: Choose an operating system that supports your chosen technologies. Linux distributions (e.g., Ubuntu, CentOS) are commonly used for web servers, while Windows Server is also an option.
   - **Web Server**: Apache, Nginx, or Microsoft IIS are popular choices for hosting web applications.
   - **Database**: Firebase
   - **Programming Languages**: Angular or React for the frontend, Node.js for the backend, and HTML/CSS for styling, Typescript
   - **Frameworks**: Express.js for building the backend of the application.
   - **Security**: Use SSL/TLS certificates for secure data transmission, and implement security best practices to protect against common vulnerabilities (e.g., SQL injection, cross-site scripting).
   - **Payment Gateway Integration**: Choose a payment gateway (e.g., PayPal, Stripe) and integrate it into your application for processing payments.
   - **Monitoring and Logging**: Use tools like New Relic, Loggly, or AWS CloudWatch for monitoring application performance and logging erro

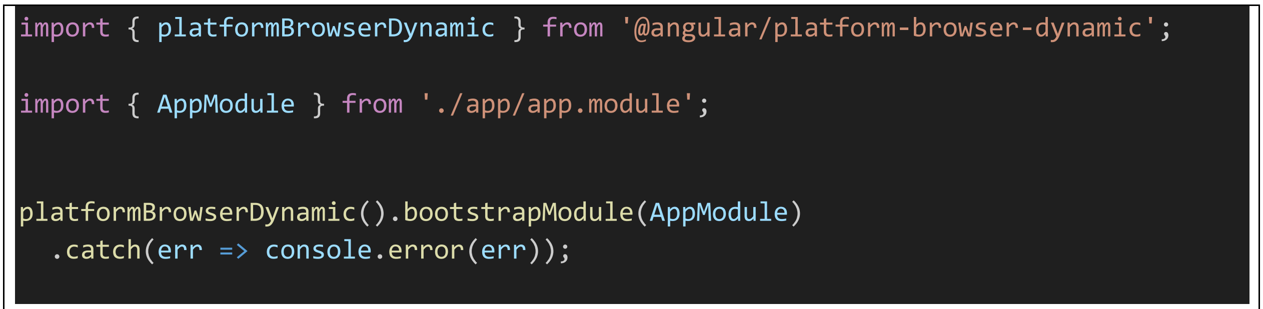
CHAPTER 3

IMPLEMENTATION

- Setup NodeJS: Install NodeJS and organize your project folders.
- Design UI: Create layouts and design using HTML, CSS, and NodeJS.
- Controllers: Develop controllers for different website sections like Recipe listings, cart, checkout, etc.
- Services: Implement services to interact with backend APIs for fetching data and managing user actions.
- State Management: Use AngularJS's data binding to manage the application state and keep UI updated.
- Backend Integration: Connect your app to backend APIs for functionalities like authentication, product listing, and cart management.
- User Authentication: Set up user login, registration, and session management.
- Recipe Cart: Develop features for adding/removing items and updating cart totals.
- Performance Optimization: Optimize your app for speed and efficiency, including script minification, image optimization, and caching.
- Testing and Deployment: Test thoroughly and deploy your app for public access.

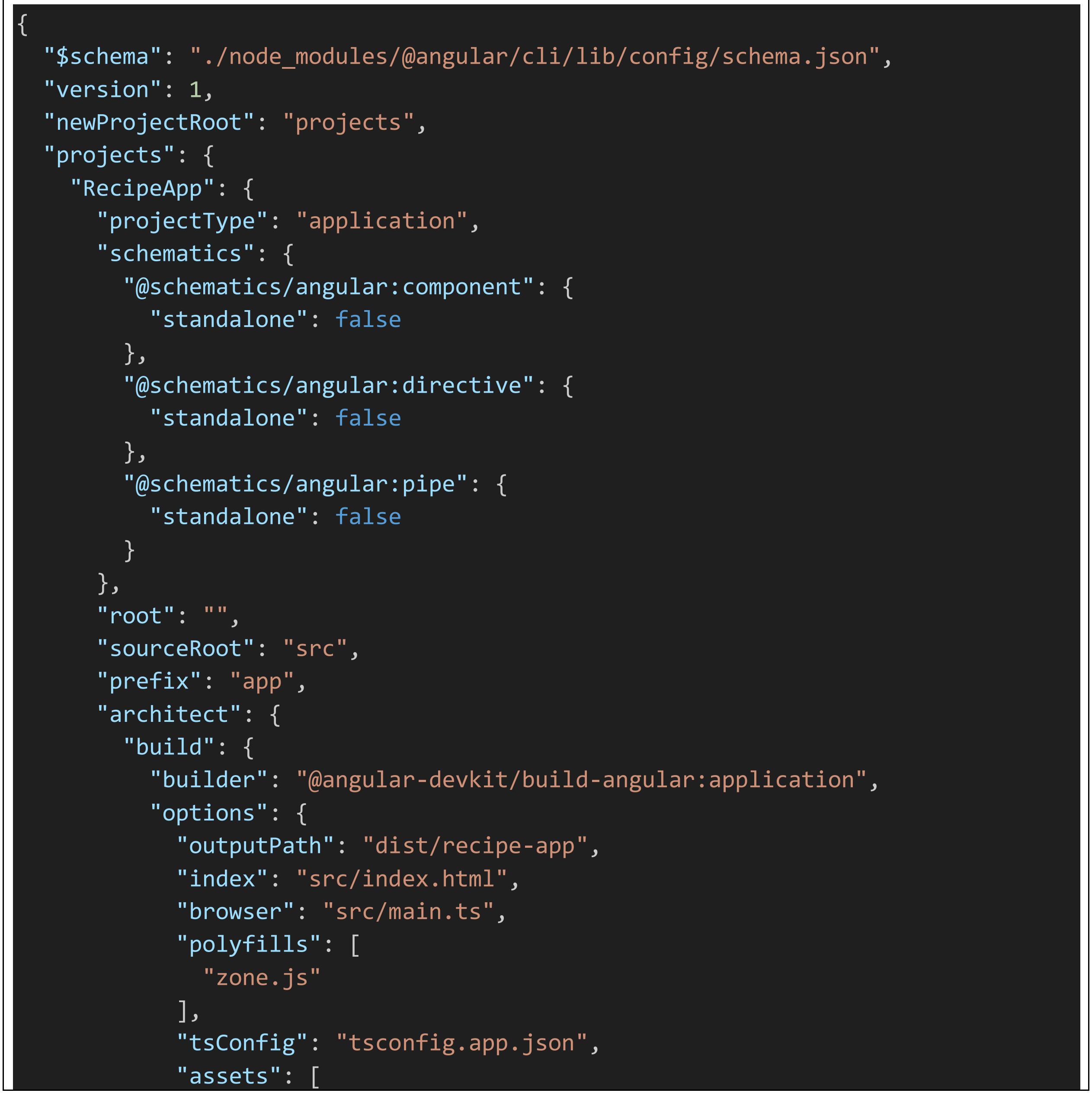
CHAPTER 4

CODE SNIPPETS

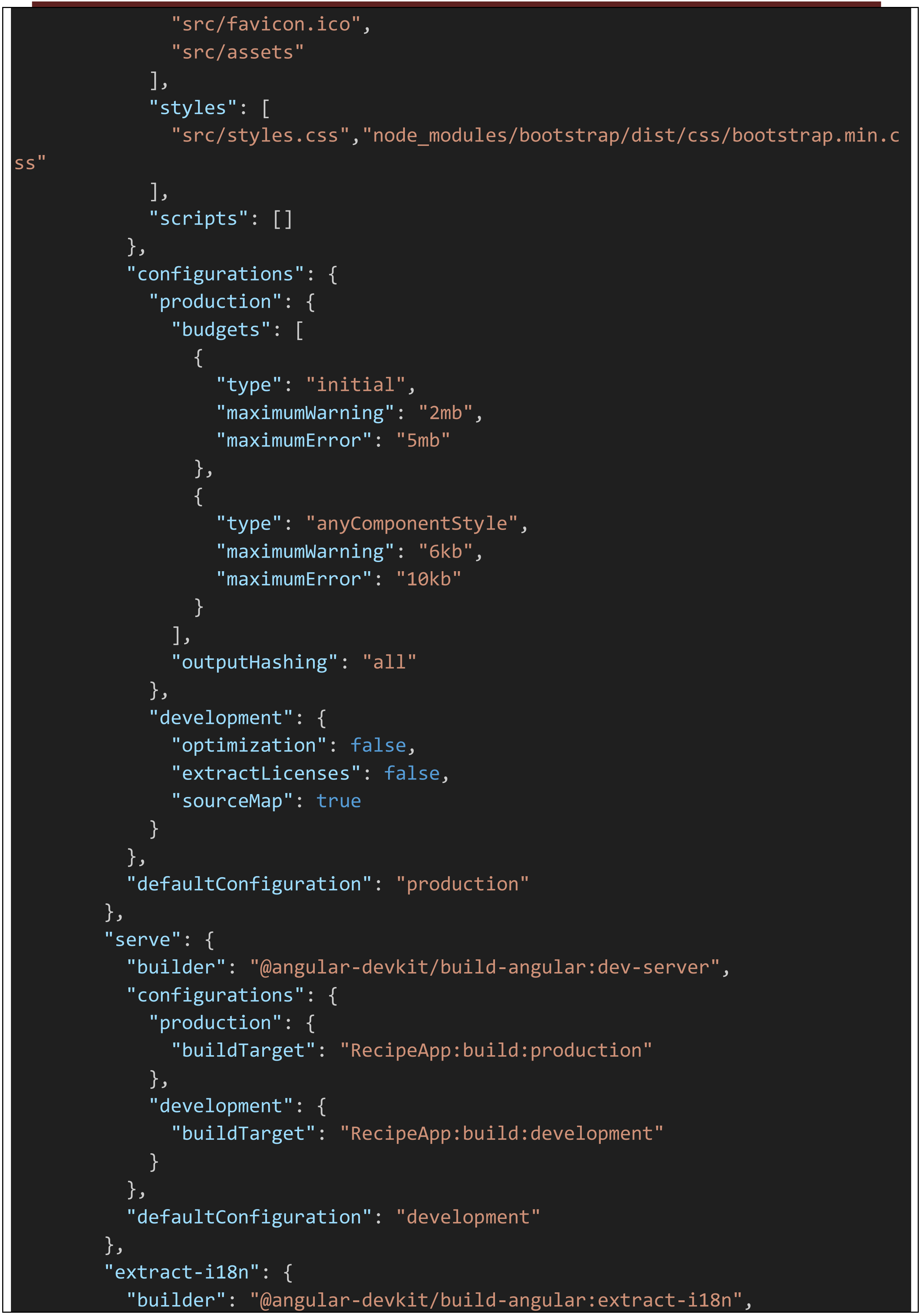


```
import { platformBrowserDynamic } from '@angular/platform-browser-dynamic';

import { AppModule } from './app/app.module';


platformBrowserDynamic().bootstrapModule(AppModule)
  .catch(err => console.error(err));
```

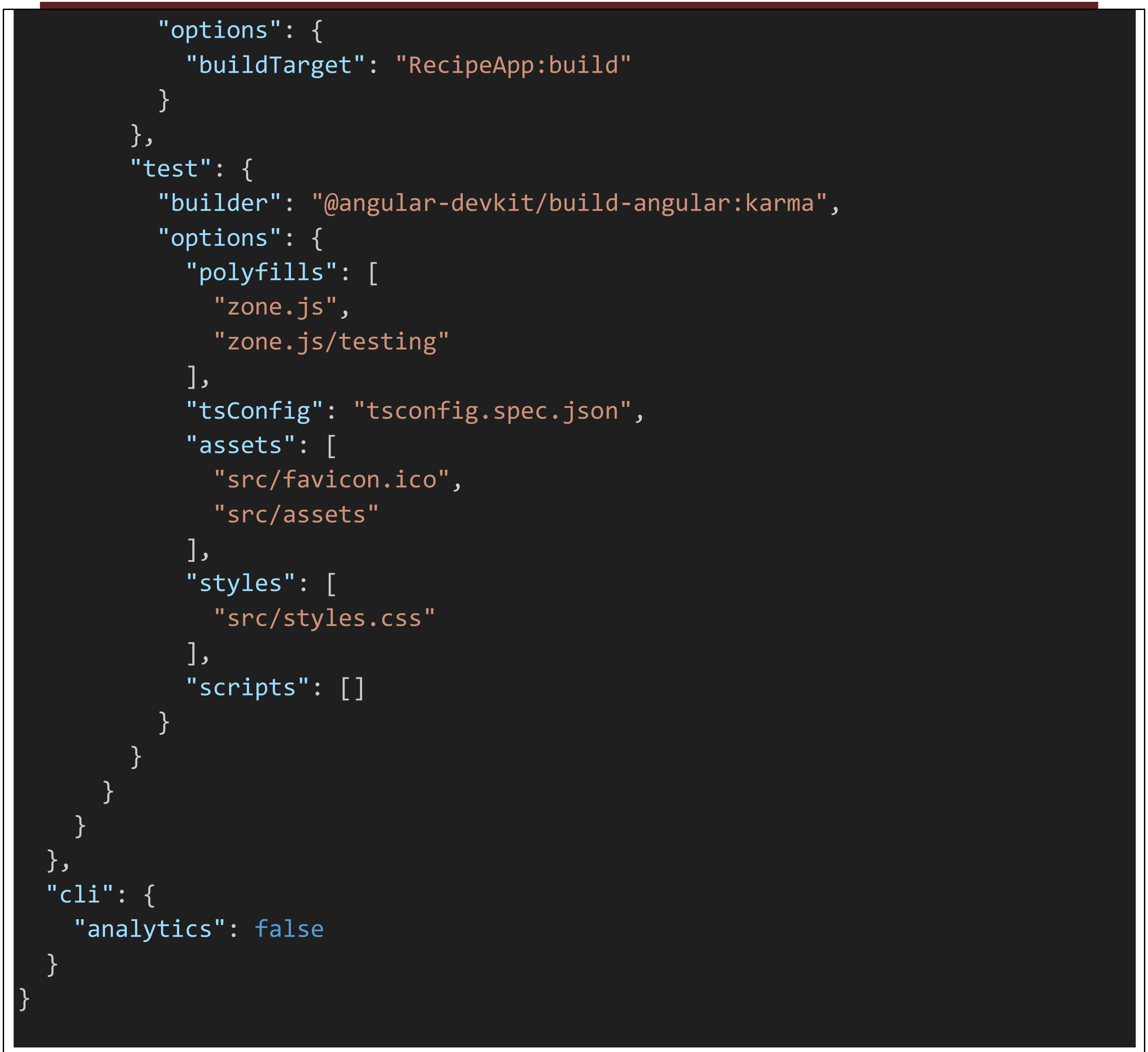
Figure 4.1



```
{
  "$schema": "./node_modules/@angular/cli/lib/config/schema.json",
  "version": 1,
  "newProjectRoot": "projects",
  "projects": {
    "RecipeApp": {
      "projectType": "application",
      "schematics": {
        "@schematics/angular:component": {
          "standalone": false
        },
        "@schematics/angular:directive": {
          "standalone": false
        },
        "@schematics/angular:pipe": {
          "standalone": false
        }
      },
      "root": "",
      "sourceRoot": "src",
      "prefix": "app",
      "architect": {
        "build": {
          "builder": "@angular-devkit/build-angular:application",
          "options": {
            "outputPath": "dist/recipe-app",
            "index": "src/index.html",
            "browser": "src/main.ts",
            "polyfills": [
              "zone.js"
            ],
            "tsConfig": "tsconfig.app.json",
            "assets": [
```

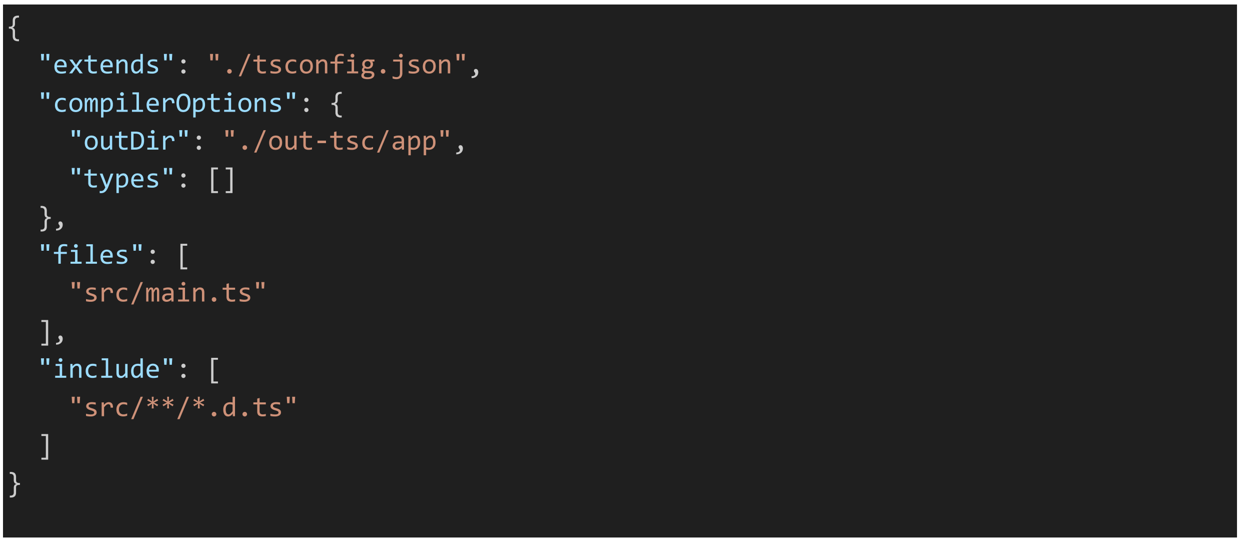


```
              "src/favicon.ico",
              "src/assets"
            ],
            "styles": [
              "src/styles.css","node_modules/bootstrap/dist/css/bootstrap.min.c
ss"
            ],
            "scripts": []
          },
          "configurations": {
            "production": {
              "budgets": [
                {
                  "type": "initial",
                  "maximumWarning": "2mb",
                  "maximumError": "5mb"
                },
                {
                  "type": "anyComponentStyle",
                  "maximumWarning": "6kb",
                  "maximumError": "10kb"
                }
              ],
              "outputHashing": "all"
            },
            "development": {
              "optimization": false,
              "extractLicenses": false,
              "sourceMap": true
            }
          },
          "defaultConfiguration": "production"
        },
        "serve": {
          "builder": "@angular-devkit/build-angular:dev-server",
          "configurations": {
            "production": {
              "buildTarget": "RecipeApp:build:production"
            },
            "development": {
              "buildTarget": "RecipeApp:build:development"
            }
          },
          "defaultConfiguration": "development"
        },
        "extract-i18n": {
          "builder": "@angular-devkit/build-angular:extract-i18n",
```



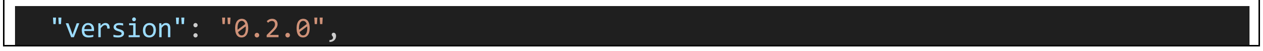
```
          "options": {
            "buildTarget": "RecipeApp:build"
          }
        },
        "test": {
          "builder": "@angular-devkit/build-angular:karma",
          "options": {
            "polyfills": [
              "zone.js",
              "zone.js/testing"
            ],
            "tsConfig": "tsconfig.spec.json",
            "assets": [
              "src/favicon.ico",
              "src/assets"
            ],
            "styles": [
              "src/styles.css"
            ],
            "scripts": []
          }
        }
      }
    }
  },
  "cli": {
    "analytics": false
  }
}
```

Figure 4.2

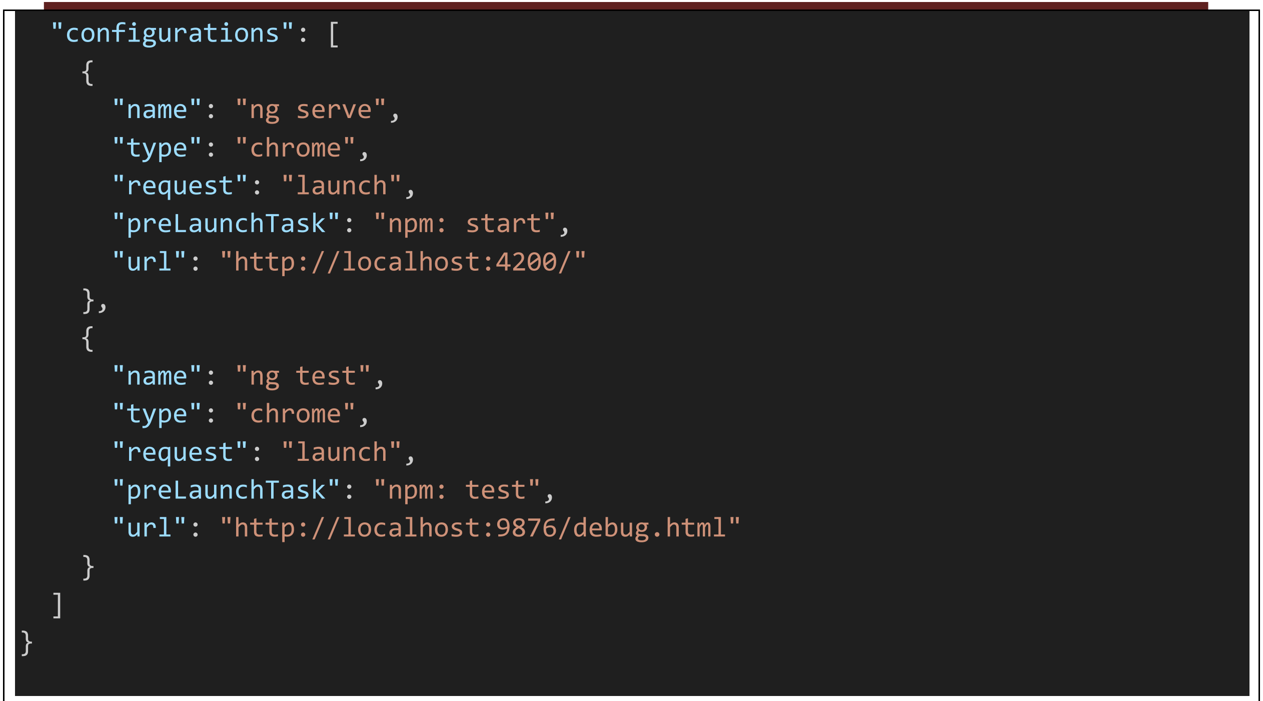


```
{
  "extends": "./tsconfig.json",
  "compilerOptions": {
    "outDir": "./out-tsc/app",
    "types": []
  },
  "files": [
    "src/main.ts"
  ],
  "include": [
    "src/**/*.d.ts"
  ]
}
```

Figure 4.3

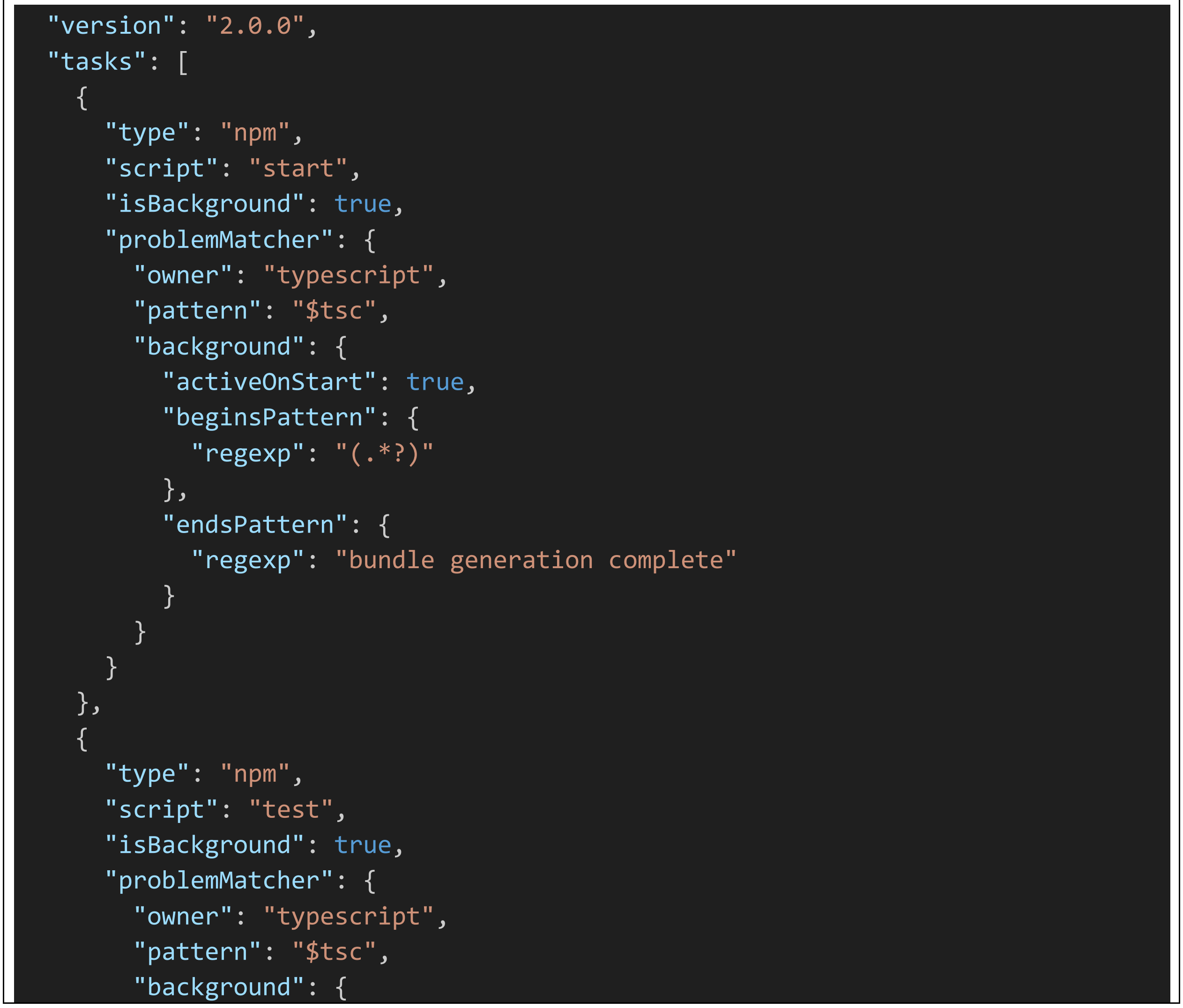


```
  "version": "0.2.0",
```

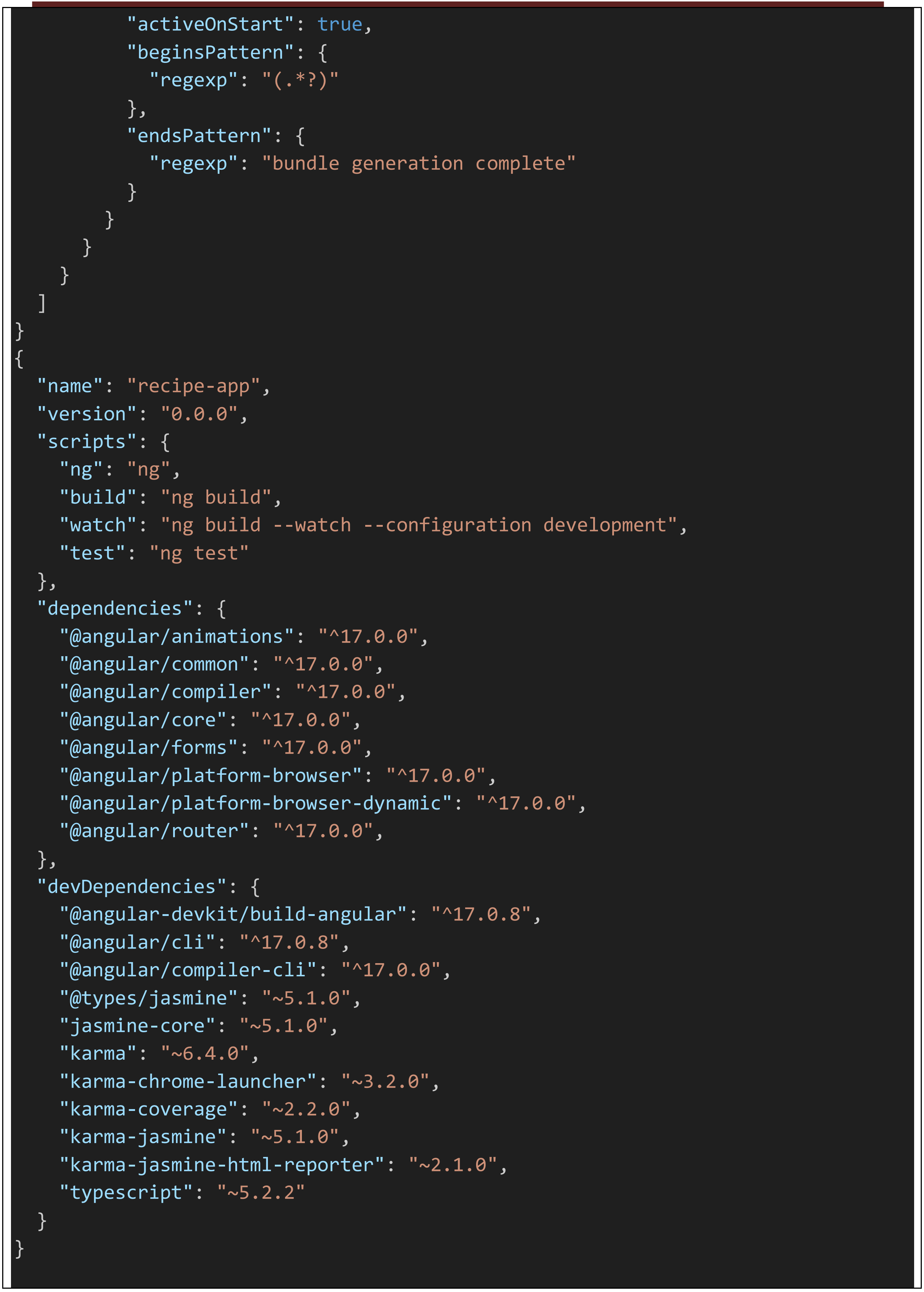


```
  "configurations": [
    {
      "name": "ng serve",
      "type": "chrome",
      "request": "launch",
      "preLaunchTask": "npm: start",
      "url": "http://localhost:4200/"
    },
    {
      "name": "ng test",
      "type": "chrome",
      "request": "launch",
      "preLaunchTask": "npm: test",
      "url": "http://localhost:9876/debug.html"
    }
  ]
}
```

Figure 4.4



```
  "version": "2.0.0",
  "tasks": [
    {
      "type": "npm",
      "script": "start",
      "isBackground": true,
      "problemMatcher": {
        "owner": "typescript",
        "pattern": "$tsc",
        "background": {
          "activeOnStart": true,
          "beginsPattern": {
            "regexp": "(.*?)"
          },
          "endsPattern": {
            "regexp": "bundle generation complete"
          }
        }
      }
    },
    {
      "type": "npm",
      "script": "test",
      "isBackground": true,
      "problemMatcher": {
        "owner": "typescript",
        "pattern": "$tsc",
        "background": {
```



```
          "activeOnStart": true,
          "beginsPattern": {
            "regexp": "(.*?)"
          },
          "endsPattern": {
            "regexp": "bundle generation complete"
          }
        }
      }
    }
  ]
}
{
  "name": "recipe-app",
  "version": "0.0.0",
  "scripts": {
    "ng": "ng",
    "build": "ng build",
    "watch": "ng build --watch --configuration development",
    "test": "ng test"
  },
  "dependencies": {
    "@angular/animations": "^17.0.0",
    "@angular/common": "^17.0.0",
    "@angular/compiler": "^17.0.0",
    "@angular/core": "^17.0.0",
    "@angular/forms": "^17.0.0",
    "@angular/platform-browser": "^17.0.0",
    "@angular/platform-browser-dynamic": "^17.0.0",
    "@angular/router": "^17.0.0",
  },
  "devDependencies": {
    "@angular-devkit/build-angular": "^17.0.8",
    "@angular/cli": "^17.0.8",
    "@angular/compiler-cli": "^17.0.0",
    "@types/jasmine": "~5.1.0",
    "jasmine-core": "~5.1.0",
    "karma": "~6.4.0",
    "karma-chrome-launcher": "~3.2.0",
    "karma-coverage": "~2.2.0",
    "karma-jasmine": "~5.1.0",
    "karma-jasmine-html-reporter": "~2.1.0",
    "typescript": "~5.2.2"
  }
}
```

CHAPTER 5

OUTPUT

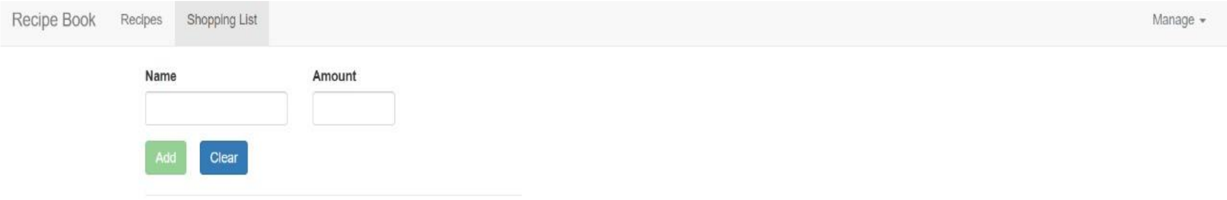
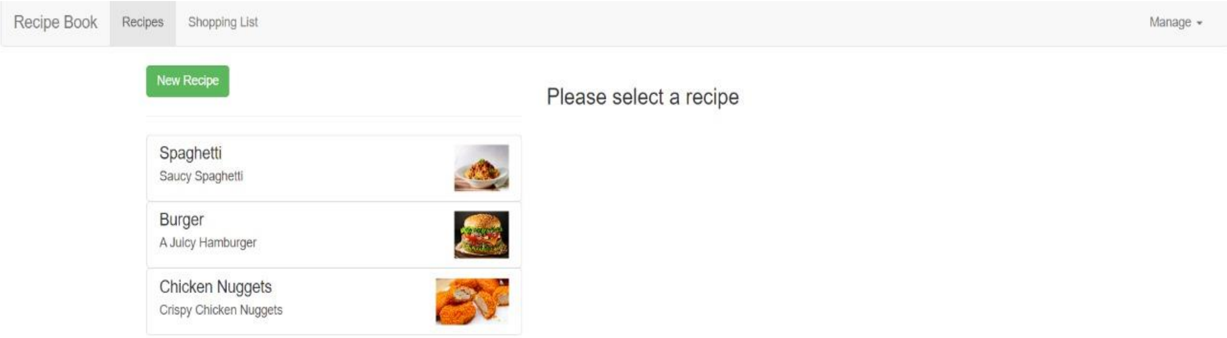


Figure 5.1

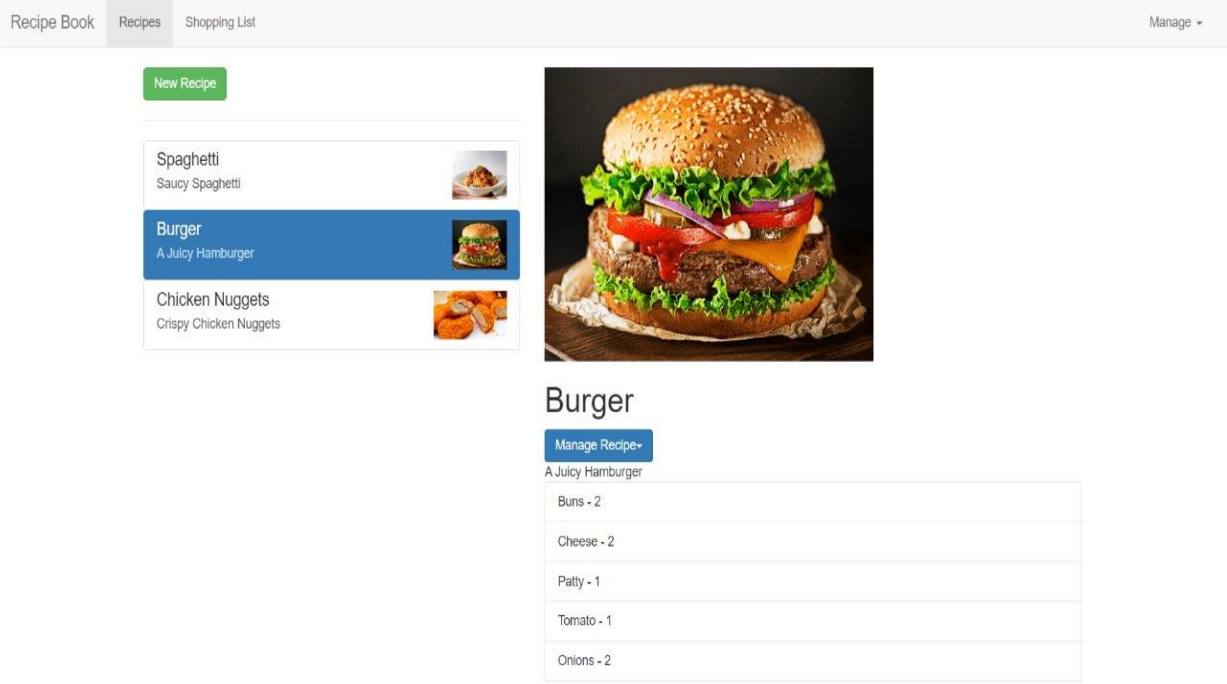


Please select a recipe

Spaghetti
Saucy Spaghetti

Burger
A Juicy Hamburger

Chicken Nuggets
Crispy Chicken Nuggets



Spaghetti
Saucy Spaghetti

Burger
A Juicy Hamburger

Chicken Nuggets
Crispy Chicken Nuggets

Burger

A Juicy Hamburger

Buns - 2

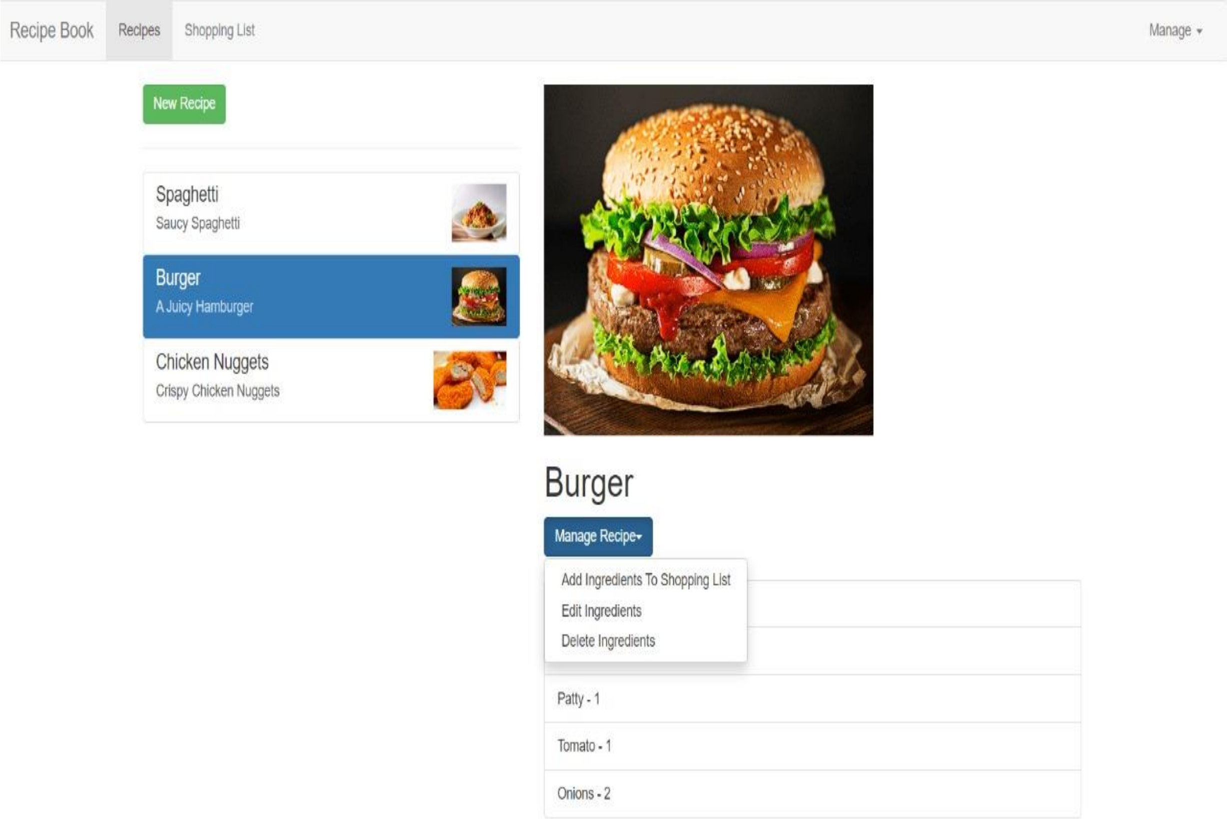
Cheese - 2

Patty - 1

Tomato - 1

Onions - 2

Figure 5.2



Spaghetti
Saucy Spaghetti

Burger
A Juicy Hamburger

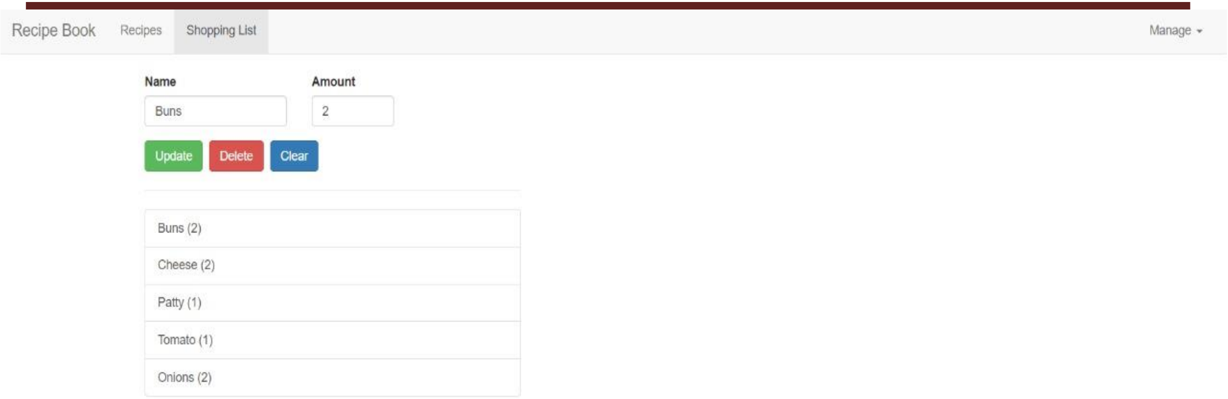
Chicken Nuggets
Crispy Chicken Nuggets

Burger

Patty - 1

Tomato - 1

Onions - 2



Name: Buns

Amount: 2

Buns (2)

Cheese (2)

Patty (1)

Tomato (1)

Onions (2)

Figure 5.3

CHAPTER 5

CONCLUSION

Conclusion

The Recipe application, built with Angular, has demonstrated robust performance and scalability, providing a seamless shopping experience for users. While current metrics reflect success, ongoing optimization efforts are necessary, particularly in leveraging Angular' s capabilities for mobile responsiveness and dynamic content delivery. Competitor analysis underscores the importance of Angular' s modular architecture for efficient feature integrationand maintenance, enabling quick adaptation to market demands.

Technological evaluation highlights Angular' s robust security features and compatibility with emerging technologies like PWAs and server-side rendering, ensuring a secure and performant application. Future growth relies on Angular 's flexibility for internationalization and localization, empowering seamless expansion into new markets. Integration of Angular Material enhances UI consistency and user engagement.

Furthermore, Angular's strong community support and frequent updates ensure the application remains at the forefront of innovation. Continued investment in Angular' s best practices, including Angular Universal for SEO optimization and NgRx for state management, will drive sustained success. Overall, Angular' s versatility and ecosystem empower the e-commerce application to thrive amidst evolving industry trends, securing its position as a leader in the competitive digital marketplace.

REFERENCES

1. https://www.google.com/

2. https://github.com/Learnshub

3. https://www.geeksforgeek.com