# Recipe App - Tài liệu Dự án

# Mục lục

- 1. Tổng quan dự án
- 2. Kiến trúc hệ thống
- 3. Backend Admin Panel (Node.js)
- 4. Backend Spring Boot (Java)
- 5. Frontend Flutter Mobile App
- 6. Cơ sở dữ liệu MongoDB
- 7. API Documentation
- 8. Cài đặt và triển khai
- 9. Tính năng chính
- 10. Cấu trúc thư mục

# Q Tổng quan dự án

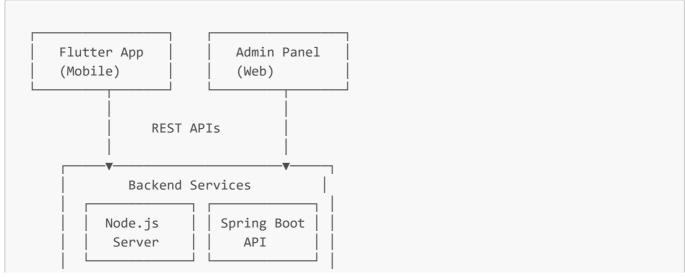
Recipe App là một ứng dụng quản lý công thức nấu ăn hoàn chỉnh bao gồm:

- Mobile App (Flutter): Úng dụng di động cho người dùng cuối
- Admin Panel (Node.js + EJS): Giao diện quản trị web
- Backend API (Spring Boot): API backend chính
- Database (MongoDB): Cơ sở dữ liệu NoSQL

## **&** Muc tiêu

- Cung cấp nền tảng chia sẻ công thức nấu ăn
- Quản lý người dùng và nội dung hiệu quả
- Hỗ trợ đa nền tảng (iOS, Android)
- Giao diện quản trị thân thiện

# **E** Kiến trúc hệ thống





## 🖔 Công nghệ sử dụng

Component	Technology Stack	
Mobile App	Flutter, Dart	
Admin Panel	Node.js, Express, EJS, Bootstrap	
Backend API	Spring Boot, Java 17	
Database	MongoDB	
Authentication	JWT, OTP Email	
File Storage	Local File System	
Email Service	Nodemailer (SMTP)	

# Backend - Admin Panel (Node.js)

## Cấu trúc thư mục

```
Script/
                       # Cấu hình
— config/
    ├─ conn.js # Kết nối MongoDB
├─ flash.js # Flash messages
└─ passport.js # Authentication
                         # Controllers
  - controllers/
    — apiController.js
    — recipeController.js
    ├─ categoryController.js
    — cuisinesController.js
      - loginController.js
  - middleware/
                         # Middleware
    — auth.js
      checkAuthentication.js
    └─ uploadMultipleFile.js
                        # Models
  - model/
    — userModel.js
    ├─ recipeModel.js
      categoryModel.js
  - routes/
                          # Routes
   views/
                         # EJS Templates
```

```
├── public/ # Static files
└── uploads/ # File uploads
```

# Tính năng chính

## 1. Quản lý người dùng

```
// controllers/apiController.js
const SignUp = async (req, res) => {
    // Đăng ký người dùng với OTP verification
    // Gửi email OTP xác thực
}

const SignIn = async (req, res) => {
    // Đăng nhập với JWT token
}
```

### 2. Quản lý công thức nấu ăn

```
// model/recipeModel.js
const recipeSchema = new mongoose.Schema({
   image: String,
    name: String,
    categoryId: ObjectId,
    cuisinesId: ObjectId,
    ingredients: [String],
    prepTime: String,
    cookTime: String,
    difficultyLevel: {
        type: String,
        enum: ["Easy", "Medium", "Hard"]
    },
    overview: String,
    how_to_cook: String
});
```

### 3. Quản lý danh mục

- Categories (Danh muc món ăn)
- Cuisines (Âm thực theo vùng miền)

## 4. Hệ thống quảng cáo

```
// model/adsModel.js
const adsSchema = new mongoose.Schema({
```

```
android_is_enable: Number,
  android_app_ad_id: String,
  android_banner_ad_id: String,
  ios_is_enable: Number,
  ios_app_ad_id: String
});
```

## **X** API Endpoints

Method	Endpoint	Mô tả
POST	/api/SignUp	Đăng ký người dùng
POST	/api/SignIn	Đăng nhập
POST	/api/VerifyOTP	Xác thực OTP
GET	/api/GetUser	Lấy thông tin user
GET	/api/getAllRecipes	Lấy tất cả công thức
POST	/api/SearchRecipes	Tìm kiếm công thức
POST	/api/AddReview	Thêm đánh giá
GET	/api/getAllCategories	Lấy danh mục
GET	/api/getAllCuisines	Lấy ẩm thực

# Backend - Spring Boot (Java)

# Cấu trúc dự án

## Models chính

### **User Model**

```
@Document(collection = "logins")
public class User {
    @Id
    private String id;
    private String firstname;
    private String lastname;
    @Indexed(unique = true)
    private String email;
    private String password;
    private String countryCode;
    private String phone;
    private Boolean isVerified = false;
    @CreatedDate
    private LocalDateTime createdAt;
    @LastModifiedDate
    private LocalDateTime updatedAt;
}
```

### **Recipe Model**

```
@Document(collection = "recipes")
public class Recipe {
    @Id
    private String id;
    private String title;
    private String description;
    private String image;
    private String categoryId;
    private String cuisineId;
    private Integer prepTime;
    private Integer cookTime;
    private Integer servings;
    private String difficulty;
    private List<String> ingredients;
    private List<String> instructions;
    private Boolean isPopular = false;
    private Boolean isRecommended = false;
    private Integer viewCount = 0;
    private Integer likeCount = 0;
}
```

# % Configuration

```
# application.properties
spring.data.mongodb.uri=mongodb://localhost:27017/recipe_db
spring.mail.host=smtp.gmail.com
spring.mail.port=587
spring.mail.username=${MAIL_USERNAME}
spring.mail.password=${MAIL_PASSWORD}
server.port=8080
```

# Frontend - Flutter Mobile App

## Cấu trúc dự án

## Tính năng UI

#### 1. Authentication Flow

- Màn hình đăng ký/đăng nhập
- Xác thực OTP qua email
- Quên mật khẩu

### 2. Home & Browse

- Hiển thị công thức phổ biến
- Tìm kiếm và lọc
- Danh muc theo cuisine

#### 3. Recipe Details

- Hiển thị chi tiết công thức
- Nguyên liệu và cách làm
- Video hướng dẫn
- Đánh giá và bình luận

### 4. User Profile

• Quản lý thông tin cá nhân

- · Công thức yêu thích
- Lich sử xem

## Dependencies chính

```
dependencies:
   flutter_localizations:
   cached_network_image:
   chewie: ^1.8.5
   connectivity_plus:
   flutter_animate: 4.5.0
   flutter_html:
   flutter_local_notifications: ^17.2.3
   http: ^1.2.2
   sqflite: ^2.4.0
   shared_preferences: ^2.3.2
```

## API Integration

```
// app_constants.dart
abstract class FFAppConstants {
   static const String imageUrl = 'http://10.0.2.2:8190/uploads/images/';
   static const String videoUrl = 'http://10.0.2.2:8190/uploads/video/';
}
```

# 🖥 Cơ sở dữ liệu MongoDB

■ Database Schema

### **Collections chính**

1. users (Người dùng)

```
{
    _id: ObjectId,
    firstname: String,
    lastname: String,
    email: String (unique),
    country_code: String,
    phone: String,
    password: String (hashed),
    avatar: String,
    isOTPVerified: Number,
    is_active: Number,
    createdAt: Date,
```

```
updatedAt: Date
}
```

## 2. recipes (Công thức nấu ăn)

```
_id: ObjectId,
 image: String,
 name: String,
 categoryId: ObjectId (ref: categories),
 cuisinesId: ObjectId (ref: cuisines),
 ingredients: [String],
 prepTime: String,
 cookTime: String,
 totalCookTime: String,
 servings: String,
 difficultyLevel: String (Easy|Medium|Hard),
 gallery: [String],
 video: String,
 overview: String,
 how_to_cook: String,
 createdAt: Date,
 updatedAt: Date
}
```

### 3. categories (Danh mục)

```
{
    _id: ObjectId,
    name: String,
    createdAt: Date,
    updatedAt: Date
}
```

## 4. cuisines (Ẩm thực)

```
{
   _id: ObjectId,
   _name: String,
   createdAt: Date,
   updatedAt: Date
}
```

## 5. **reviews** (Đánh giá)

```
{
   _id: ObjectId,
   userId: ObjectId (ref: users),
   recipeId: ObjectId (ref: recipes),
   rating: Number,
   comment: String,
   createdAt: Date,
   updatedAt: Date
}
```

## 6. favouriteRecipes (Công thức yêu thích)

```
{
   __id: ObjectId,
   userId: ObjectId (ref: users),
   recipeId: ObjectId (ref: recipes),
   createdAt: Date,
   updatedAt: Date
}
```

## 7. otps (OTP verification)

```
{
   _id: ObjectId,
   userId: ObjectId (ref: users),
   email: String,
   otp: Number,
   createdAt: Date,
   updatedAt: Date
}
```

#### 8. intro (Intro slides)

```
{
   _id: ObjectId,
   image: String,
   ittle: String,
   description: String,
   createdAt: Date,
   updatedAt: Date
}
```

### 9. ads (Quảng cáo)

```
{
    _id: ObjectId,
    android_is_enable: Number,
    android_app_ad_id: String,
    android_banner_ad_id: String,
    ios_is_enable: Number,
    ios_app_ad_id: String,
    // ... more ad fields
    createdAt: Date,
    updatedAt: Date
}
```

### 10. faqs (FAQ)

```
{
   _id: ObjectId,
   question: String,
   answer: String,
   status: String (Publish|UnPublish),
   createdAt: Date,
   updatedAt: Date
}
```

## Relationships

```
erDiagram

USERS ||--o{ REVIEWS : creates

USERS ||--o{ FAVOURITE_RECIPES : has

RECIPES ||--o{ REVIEWS : receives

RECIPES ||--o{ FAVOURITE_RECIPES : liked_by

RECIPES }o--|| CATEGORIES : belongs_to

RECIPES }o--|| CUISINES : belongs_to

USERS ||--o{ OTPS : verified_by
```

# API Documentation

Authentication APIs

## Đăng ký người dùng

```
POST /api/SignUp
Content-Type: application/json
{
    "firstname": "John",
```

```
"lastname": "Doe",
   "email": "john@example.com",
   "country_code": "+84",
   "phone": "123456789",
   "password": "password123"
}
```

### **Response:**

```
{
   "status": true,
   "message": "OTP sent to your email",
   "userId": "64a1b2c3d4e5f6789abcdef0"
}
```

## Xác thực OTP

```
POST /api/VerifyOTP
Content-Type: application/json

{
   "userId": "64a1b2c3d4e5f6789abcdef0",
   "otp": 1234
}
```

#### Đăng nhập

```
POST /api/SignIn
Content-Type: application/json

{
    "email": "john@example.com",
    "password": "password123"
}
```

### **Response:**

```
{
  "status": true,
  "message": "Login successful",
  "data": {
    "user": { /* user data */ },
    "token": "jwt_token_here"
```

```
}
```

Recipe APIs

## Lấy tất cả công thức

```
GET /api/getAllRecipes
Authorization: Bearer <jwt_token>
```

### Tìm kiếm công thức

```
POST /api/SearchRecipes
Authorization: Bearer <jwt_token>
Content-Type: application/json

{
    "search": "chicken",
    "categoryId": "optional_category_id",
    "cuisineId": "optional_cuisine_id"
}
```

### Thêm đánh giá

```
POST /api/AddReview
Authorization: Bearer <jwt_token>
Content-Type: application/json

{
    "recipeId": "64a1b2c3d4e5f6789abcdef0",
    "rating": 5,
    "comment": "Delicious recipe!"
}
```

## Category & Cuisine APIs

## Lấy danh mục

```
GET /api/getAllCategories
```

### Lấy ẩm thực

```
GET /api/getAllCuisines
```

## Favorite APIs

## Thêm/xóa yêu thích

```
POST /api/AddFavouriteRecipe
Authorization: Bearer <jwt_token>
Content-Type: application/json

{
    "recipeId": "64a1b2c3d4e5f6789abcdef0"
}
```

# Cài đặt và triển khai

% Yêu cầu hệ thống

Node.js: v16+
Java: 17+
Flutter: 3.5+
MongoDB: 4.0+
Maven: 3.6+

Cài đặt Backend (Node.js)

## 1. Clone repository và cài đặt dependencies:

```
cd "Recipe App Admin Panel Source Code/Script"
npm install
```

## 2. Cấu hình environment variables:

```
# .env
DB_URL=mongodb://localhost:27017/recipe_db
JWT_SECRET=your_jwt_secret
MAIL_HOST=smtp.gmail.com
MAIL_PORT=587
MAIL_USER=your_email@gmail.com
MAIL_PASS=your_app_password
IMAGE_URL=http://localhost:8190/uploads/images/
VIDEO_URL=http://localhost:8190/uploads/video/
```

#### 3. Chay server:

```
npm start
# hoặc
nodemon index.js
```

## Cài đặt Backend (Spring Boot)

#### 1. Cấu hình database:

```
# src/main/resources/application.properties
spring.data.mongodb.uri=mongodb://localhost:27017/recipe_db
spring.mail.username=your-email@gmail.com
spring.mail.password=your-app-password
```

### 2. Build và chạy:

```
cd backend
mvn clean install
mvn spring-boot:run
```

## Cài đặt Flutter App

### 1. Cài đặt dependencies:

```
cd "Flutter Receipe App/recipe_app"
flutter pub get
```

⚠ Lưu ý: Nếu gặp lỗi dependency conflict với intl, hãy cập nhật trong pubspec.yaml:

```
# Thay đổi từ:
intl: 0.19.0
# Thành:
intl: ^0.20.2
```

### 2. Cấu hình API endpoints:

```
// lib/app_constants.dart
abstract class FFAppConstants {
   static const String imageUrl = 'http://your-server:8190/uploads/images/';
```

```
static const String videoUrl = 'http://your-server:8190/uploads/video/';
}
```

### 3. Chạy app:

```
flutter run
```

## 🖁 Cài đặt MongoDB

### 1. Khởi tạo database:

```
mongod --dbpath /path/to/data
```

## 2. Import dữ liệu mẫu:

```
mongoimport --db recipe_db --collection categories --file categories.json
mongoimport --db recipe_db --collection cuisines --file cuisines.json
mongoimport --db recipe_db --collection recipes --file recipes.json
```

# Tính năng chính

# **1** Quản lý người dùng

### Đăng ký & Xác thực

- Dăng ký với email và số điện thoại
- ✓ Xác thực OTP qua email
- Dăng nhập với JWT token
- Quên mât khẩu với OTP
- Quản lý profile người dùng

## Phân quyền

- User thường (xem, đánh giá, yêu thích)
- Admin (quản lý toàn bộ hệ thống)

## O Quản lý công thức nấu ăn

## **CRUD Operations**

- ☑ Thêm/sửa/xóa công thức
- Upload hình ảnh và video
- Quản lý nguyên liệu và cách làm

• Phân loại theo danh mục và ẩm thực

## Tính năng nâng cao

- 🗹 Tìm kiếm và lọc công thức
- Dánh giá và bình luận
- ✓ Yêu thích công thức
- 🗹 Gợi ý công thức phổ biến
- ☑ Thống kê lượt xem

## Quản lý danh mục

## Categories (Danh mục món ăn)

- Món chính
- Món tráng miệng
- Đồ uống
- Món ăn vặt

## Cuisines (Âm thực theo vùng)

- Ẩm thực Việt Nam
- Âm thực Trung Hoa
- Ẩm thực Nhật Bản
- Ẩm thực Âu

# **Ⅲ** Hệ thống quản trị

### **Dashboard**

- ✓ Thống kê người dùng
- ✓ Thống kê công thức
- 🗹 Thống kê đánh giá
- ☑ Biểu đồ analytics

## Quản lý nội dung

- Quản lý intro slides
- Quản lý FAQ
- 🗹 Cài đặt quảng cáo
- 🗹 Cài đặt hệ thống

## **Mobile App Features**

### UI/UX

- Responsive design
- ☑ Dark/Light theme
- Smooth animations

• Offline caching

### Tính năng đặc biệt

- Video streaming
- **Push notifications**
- Share recipes
- ✓ Multi-language support

## **Monetization**

## Hệ thống quảng cáo

- 🗹 Banner ads
- Interstitial ads
- Rewarded ads

### **Analytics**

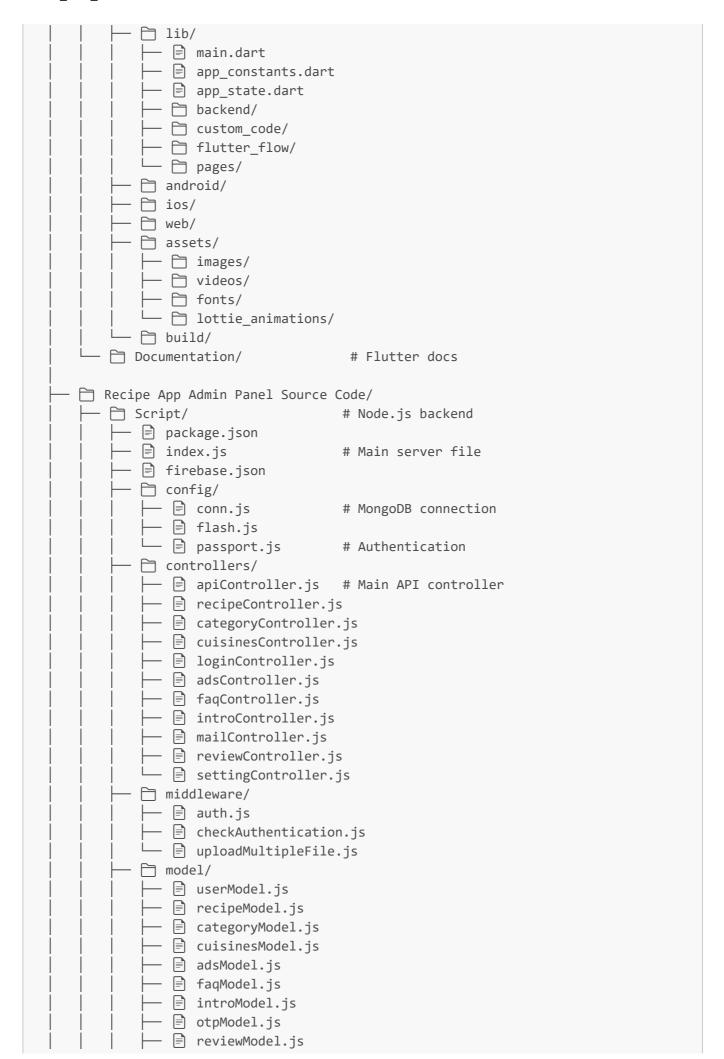
- User engagement tracking
- Recipe popularity metrics
- Ad performance monitoring

# Cấu trúc thư mục chi tiết

```
RecipeAPPAI2/
 — 🖹 README.md
 —    recipe-app.code-workspace
 — 🖹 test_api.js
                                  # API testing script
 ─ ☐ LOGIN_FIX_SUMMARY.md
                                 # Login fix documentation
  - 🗐 fix api calls.ps1
                                  # API fix script
  – 🖹 fix_api_parsing.ps1
                                  # API parsing fix script

    backend/

                                    # Spring Boot backend
    ├─ 📄 pom.xml
      − 🖹 README.md
      – 🗎 src/main/java/com/recipe/admin/
        ─ ☐ controller/
         ─ ☐ model/
       ├─ 🗎 repository/
          - 🗎 service/
         – 🖹 RecipeAdminApplication.java
      - 🗀 target/
                                  # Compiled files
   Flutter Receipe App/
                             # Flutter mobile app
     — 🗎 recipe_app/
         − 🗐 pubspec.yaml
           analysis_options.yaml
```





# Troubleshooting

Common Issues

### 1. Flutter Dependencies Conflict

Error: Because recipe\_app depends on flutter\_localizations from sdk which depends on intl 0.20.2, intl 0.20.2 is required.

#### **Solution:**

```
# Cập nhật intl version trong pubspec.yaml
intl: ^0.20.2

# Sau đó chạy:
flutter pub get
```

### 2. MongoDB Connection Issues

```
Error: MongoNetworkError: connect ECONNREFUSED
```

#### **Solution:**

```
# Kiểm tra MongoDB service

mongod --version

# Khởi động MongoDB

mongod --dbpath /path/to/data
```

### 3. Node.js Port Already in Use

```
Error: listen EADDRINUSE: address already in use :::8190
```

### **Solution:**

#### 4. Flutter Build Issues

```
Error: Gradle build failed
```

#### **Solution:**

```
# Clean build
flutter clean
flutter pub get
# Rebuild
flutter run
```

### 5. API Connection from Flutter

**Problem:** Cannot connect to localhost API from mobile app

#### **Solution:**

```
// Thay đổi localhost thành IP thực tế trong app_constants.dart
static const String imageUrl = 'http://192.168.1.100:8190/uploads/images/';
// Hoặc sử dụng Android emulator localhost:
static const String imageUrl = 'http://10.0.2.2:8190/uploads/images/';
```

## Debug Commands

```
# Check Flutter doctor
flutter doctor
# Check dependencies
flutter pub deps
# Check outdated packages
flutter pub outdated
# Analyze code
flutter analyze
# Run tests
flutter test
# Check MongoDB connection
mongo --eval "db.runCommand('ping')"
# Check Node.js server logs
npm run start
# Check Spring Boot health
curl http://localhost:8080/actuator/health
```

# 🚱 🗗 Bảo mật

- Authentication & Authorization
  - **JWT Token**: Xác thực API với thời gian hết hạn
  - Password Hashing: SHA256 double hash
  - OTP Verification: Email OTP cho đăng ký/reset password
  - Role-based Access: Phân quyền admin/user

## ெ Data Protection

- Input Validation: Kiểm tra dữ liệu đầu vào
- SQL Injection Prevention: Sử dụng Mongoose ODM
- XSS Protection: Escape HTML content

• CORS Configuration: Cấu hình CORS phù hợp

## 

- API Rate Limiting: Giới hạn request per minute
- OTP Rate Limiting: Giới hạn gửi OTP
- File Upload Limits: Giới hạn kích thước file

# Performance & Monitoring

# Optimization

- Database Indexing: Index cho email, categoryld, cuisineld
- Image Optimization: Resize và compress images
- Caching: Cache API responses
- Pagination: Phân trang cho danh sách lớn

## Monitoring

- Error Logging: Winston logger
- API Analytics: Request/response tracking
- Database Monitoring: MongoDB metrics
- User Analytics: App usage statistics

# Testing

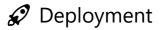
# ≜ Test Scripts

```
# API Testing
node test_api.js
node test_addreview.js
node test_getrecipe.js
node test_cuisine_api.js

# Database Testing
node check_all_recipes_cuisines.js
node list_cuisines.js
```

# Test Coverage

- Unit Tests: Model validation
- Integration Tests: API endpoints
- End-to-End Tests: Full user workflows
- Performance Tests: Load testing



## Production Environment

### **Server Requirements**

CPU: 2+ coresRAM: 4GB+

Storage: 50GB+ SSDOS: Ubuntu 20.04+

## **Database Setup**

```
# MongoDB Production
mongodb://username:password@host:27017/recipe_db?authSource=admin
```

#### **Environment Variables**

```
# Production .env
NODE_ENV=production
DB_URL=mongodb://prod-server:27017/recipe_db
JWT_SECRET=strong_production_secret
MAIL_HOST=smtp.gmail.com
API_BASE_URL=https://api.recipeapp.com
ADMIN_URL=https://admin.recipeapp.com
```

# Mobile App Release

### **Android Build**

```
flutter build appbundle --release
```

#### iOS Build

```
flutter build ios --release
```

# Future Enhancements

Roadmap

## **Phase 1: Core Improvements**

• Real-time notifications

- Advanced search filters
- Recipe recommendations AI
- Social sharing integration

#### **Phase 2: Advanced Features**

- Video calling for cooking classes
- AR recipe visualization
- Voice-guided cooking
- Meal planning calendar

#### **Phase 3: Business Features**

- Subscription model
- Premium recipes
- Chef partnerships
- E-commerce integration

## **\*** Technical Improvements

- Microservices architecture
- Redis caching
- CDN integration
- Auto-scaling
- GraphQL API
- Real-time chat
- Progressive Web App

# **Support & Contact**

# SOS Technical Support

- Email: support@recipeapp.com
- **Documentation**: [Link to detailed docs]
- Issue Tracker: [GitHub Issues]
- Community: [Discord/Slack]

## ■ Development Team

- Project Manager: [Name]
- Backend Developer: [Name]
- Mobile Developer: [Name]
- **UI/UX Designer**: [Name]



Copyright (c) 2024 Recipe App Team All rights reserved.

This software and associated documentation files (the "Software") are proprietary and confidential. Unauthorized copying, distribution, or modification is prohibited.



# Acknowledgments

- Flutter Team cho framework tuyệt vời
- MongoDB cho database linh hoạt
- Node.js Community cho ecosystem phong phú
- Spring Boot cho backend framework manh me

Tài liệu này được tạo vào ngày 4 tháng 9, 2025 và sẽ được cập nhật thường xuyên.