



PICK MY DISH

Your Personal Cooking Companion

- Time-aware
- Mood-based
- Ingredient-aware

Intelligent Recipe Recommendation System

Team Members:

Kamdeu Yamdjeuson Neil Marshall (Backend & DevOps)
Tuheu Tchoubi Pempeme Moussa Fahdil (Frontend & UI/UX)



THE CHALLENGE



Decision Fatigue



Limited Resources



Time Constraints



No Personalization



OUR SOLUTION



AI Recommendations



Smart Filtering



Quick Planning



Mood-Aware

Intelligent Features for Seamless Cooking



**Mood
Filtering**



**Smart Ingredient
Matching**



**Time-Aware
Planning**



**Recipe
Sharing**



**Admin
Dashboard**



**Personal
Favorites**



**Offline
Access**



**User
Profiles**

Modern Stack for Scalable Solutions

Frontend



Flutter



Provider State



Material UI

Backend



Node.js



Express.js



MySQL



JWT + Bcrypt



DevOps & Tools



Git/GitHub



Jenkins

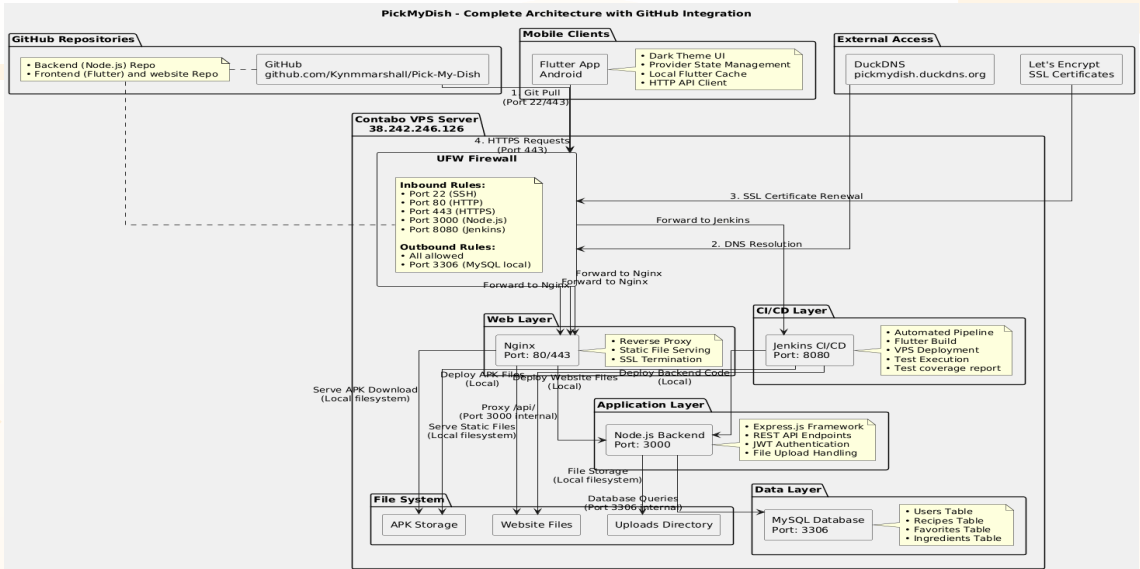


Nginx

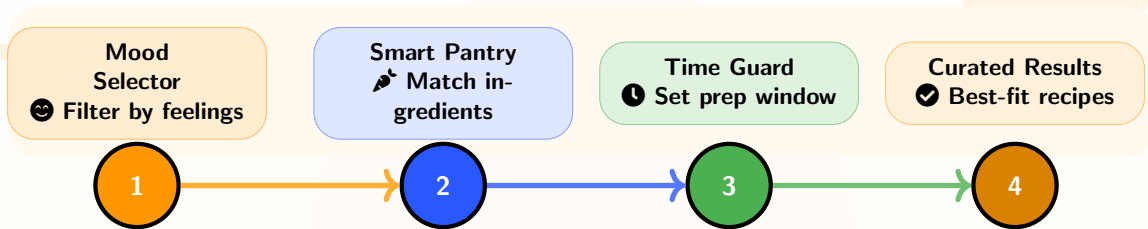


Contabo VPS

System Architecture



User Flow: Mood-Based Recipe Discovery [INNOVATIONS]



Happy • Calm • Energetic • Comforting • Healthy • Quick • Light

Why it impresses: Adaptive filtering blends mood, pantry items, and prep time to present a shortlist that feels tailored. Real-time scoring ranks recipes by relevance, freshness, and dietary fit.

Design Patterns



Provider Screens rebuild from app state



Repository APIs + cache behind one interface



Factory Creates recipe/user models consistently



Singleton One config/auth client shared



Observer Mood/ingredient changes trigger UI

SOLID in Practice

S — Single Responsibility

UI shows, repo fetches, service computes

O — Open/Closed Add filters without changing base screens

L — Liskov Substitution

Swap any recipe source via the interface

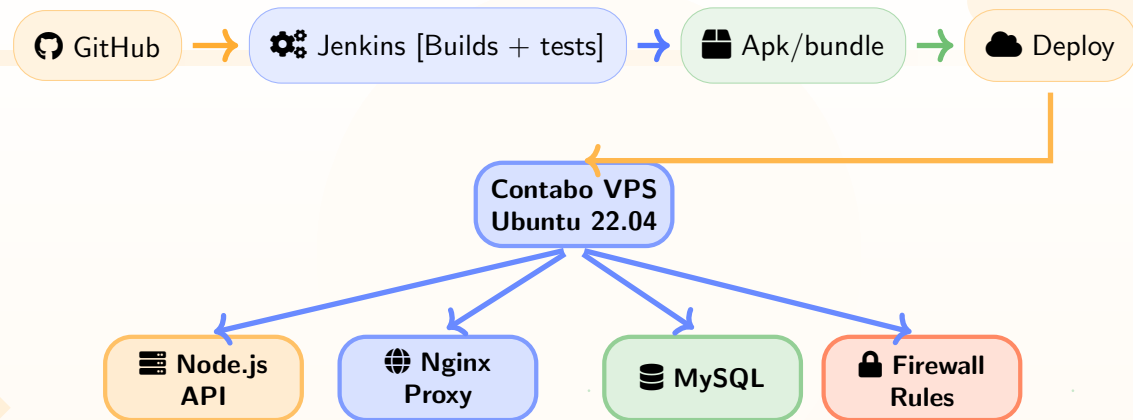
I — Interface Segregation

Separate auth, recipe, storage

D — Dependency Inversion

UI leans on abstractions; inject repos

Deployment & DevOps Pipeline (SUMMARY)



🕒 CI = 5 minutes — 🔄 Zero-downtime proxy swaps — 📈 Monitored uptime 99.8%

Coverage (Actual)

80.9% actual

80% target

Lowest

coverage (focus next):

profile_screen.dart 65.1%

recipe_upload_screen.dart 65.6%

recipe_edit_screen.dart 68.5%

Performance (Latest Run)

Metric	Target	Actual	
Recipe Load Time	2s	1.6s	
API Response	500ms	410ms	
DB Query	300ms	240ms	
Uptime (30d)	99.8%	99.6%	

Optimizations:

- Redis caching for frequent recipes
 - API response optimization
- Real-time monitoring alerts

Achievements

- Fully functional cross-platform app
- Secure authentication system
- Seamless data synchronization across devices
- Fast, responsive UI
- Automated deployment
- Comprehensive testing


Future Enhancements

- AI-powered recommendations
- Multi-language support
- Advanced analytics
- Smart notifications
- Nutrition tracking
- Social sharing

Thank You!

Questions & Discussion



 [GitHub Repository](#)

 [Live Application](#)

Scan to Access



`pickmydish.duckdns.org`

Mobile app APK available for download