



15-Day Development Plan

Phase 1: Backend & Auth (Days 1–3)

Day 1

- Initialize Node.js + Express project
- Setup MongoDB connection (Mongoose)
- Configure dotenv for environment variables
- Create User model (email, password)

Day 2

- Implement JWT authentication (login, register routes)
- Password hashing with bcrypt
- Setup middleware for protected routes

Day 3

- Build simple API routes for testing auth (e.g., /me)
 - Postman tests for login/register
 - Setup basic error handling & validation (express-validator)
-

Phase 2: GitHub Webhook Integration (Days 4–5)

Day 4

- Implement /webhooks/github endpoint
- Verify HMAC SHA-256 signature (crypto)
- Log received PR events

Day 5

- Fetch changed files from GitHub API (GET /pulls/{id}/files)
 - Store PR metadata in MongoDB (repo, PR number, author)
-

Phase 3: AI Service (Python) (Days 6–8)

Day 6

- Initialize FastAPI project
- Create /analyze endpoint
- Define Pydantic model for PR diff input/output JSON

Day 7

- Implement LangChain + LangGraph orchestration
 - Agent 1: Lint & Style
 - Agent 2: Bug detection
 - Agent 3: Security scan
 - Agent 4: Performance review

Day 8

- Add Coordinator Agent (merge results into final JSON)
 - Return structured JSON (score, categories, comments, fixes)
 - Test with mock PR data
-

Phase 4: Node ↔ Python Communication (Day 9)

Day 9

- From Node backend → send PR diffs to Python API (axios POST)
 - Receive JSON response → save to MongoDB
 - Handle errors (timeout, invalid response)
-

Phase 5: GitHub Review Posting (Day 10)

Day 10

- Use GitHub API: POST review comments back to PR
 - Post both inline comments & overall summary
 - Verify on a test repo
-

Phase 6: Frontend (Days 11–13)

Day 11

- Setup React project (Vite or CRA)
- Implement Login/Register pages (JWT auth)
- Store JWT in localStorage or httpOnly cookie

Day 12

- Dashboard: list all PR reviews (title, repo, score, date)
- Connect API → fetch reviews for logged-in user

Day 13

- Review detail page: show breakdown (lint/bugs/security/performance)
 - Display inline comments + fix suggestions
 - (Optional) Add score chart (Chart.js/Recharts)
-

Phase 7: Testing & Deployment (Days 14–15)

Day 14

- Test full flow with real GitHub PR
- Debug webhook triggers, API calls, and AI responses
- Add logging & error handling

Day 15

- Write README (setup instructions, API docs)
 - Add .env.example
 - Final cleanup & polish
 - Deploy (Render/Heroku for Node + Python, MongoDB Atlas, Vercel/Netlify for frontend)
-

✅ By Day 15, you'll have a **working end-to-end system**:

- GitHub PRs → Webhook → Node → Python AI → GitHub review comments → React dashboard