



# CareerPilot Development Roadmap

**Project Duration:** 4 Weeks

**Start Date:** November 18, 2025

**Target Completion:** December 16, 2025

---



## Table of Contents

- [Overview](#)
  - [Week 1: Foundation & Setup](#)
  - [Week 2: Core Features](#)
  - [Week 3: AI Integration](#)
  - [Week 4: Polish & Deploy](#)
  - [Daily Time Commitment](#)
  - [Success Metrics](#)
  - [Risk Mitigation](#)
- 



## Overview

This 4-week intensive roadmap will take you from empty folders to a fully functional, deployable AI-powered career advisor platform. Each week builds upon the previous, with clear milestones and deliverables.

### Development Philosophy:

- ☒ Build iteratively, test frequently
  - ☒ Complete features end-to-end before moving on
  - ☒ Commit daily, document as you go
  - ☒ Focus on MVP, note future enhancements
- 



## Week 1: Foundation & Setup

**Goal:** Establish project foundation with authentication and database

### Day 1 (Nov 18) - Database & Models 🕒 4-5 hours

#### Morning:

- ☐ Set up PostgreSQL database locally
- ☐ Configure Sequelize in backend
- ☐ Create database configuration file

#### Afternoon:

- ☐ Create all Sequelize models (Users, Profiles, Skills, etc.)
- ☐ Write migration files
- ☐ Test migrations (up and down)

#### Evening:

- ☐ Create seed data for testing
- ☐ Run seeders

- ☐ Verify database structure

**Deliverables:**

- ☒ Working PostgreSQL database
  - ☒ All tables created with relationships
  - ☒ Sample data seeded
- 

**Day 2 (Nov 19) - Backend Setup & Auth 🕒 5-6 hours**

**Morning:**

- ☐ Set up Express server ( `app.js` , `server.js` )
- ☐ Configure middleware (CORS, body-parser, cookie-parser)
- ☐ Create error handling middleware
- ☐ Set up environment variables ( `.env` )

**Afternoon:**

- ☐ Build authentication system:
  - ☐ Registration endpoint
  - ☐ Login endpoint
  - ☐ JWT token generation
  - ☐ Password hashing (bcrypt)

**Evening:**

- ☐ Create auth middleware for protected routes
- ☐ Test auth endpoints with Postman/Thunder Client
- ☐ Write input validation

**Deliverables:**

- ☒ Express server running on port 5000
  - ☒ Working registration & login
  - ☒ JWT authentication functional
- 

**Day 3 (Nov 20) - Frontend Setup & Auth UI 🕒 5-6 hours**

**Morning:**

- ☐ Set up React Router
- ☐ Create base layout components (Navbar, Footer)
- ☐ Set up Axios configuration
- ☐ Create authentication context

**Afternoon:**

- ☐ Build Login page
- ☐ Build Registration page
- ☐ Create form validation
- ☐ Style with TailwindCSS

**Evening:**

- ☐ Implement protected route wrapper
- ☐ Connect login/register forms to backend
- ☐ Test authentication flow
- ☐ Store JWT in localStorage

**Deliverables:**

- ☒ Login and Register pages working
  - ☒ Token stored after successful login
  - ☒ Protected routes functional
- 

**Day 4 (Nov 21) - User Profile System 🕒 5-6 hours****Morning:**

- ☐ Backend: Profile CRUD endpoints
  - ☐ GET user profile
  - ☐ UPDATE user profile
- ☐ Create profile service logic

**Afternoon:**

- ☐ Frontend: Profile page UI
- ☐ Profile editor component
- ☐ Skills selector component
- ☐ Interests multi-select

**Evening:**

- ☐ Connect profile page to backend
- ☐ Test profile updates
- ☐ Add loading states and error handling

**Deliverables:**

- ☒ User can view and edit profile
  - ☒ Skills and interests management
  - ☒ Profile data persists to database
- 

**Day 5 (Nov 22) - Dashboard Foundation 🕒 4-5 hours****Morning:**

- ☐ Create dashboard page layout
- ☐ Build stats cards component
- ☐ Create progress bar component
- ☐ Design dashboard grid layout

**Afternoon:**

- ☐ Backend: Dashboard stats endpoint

- ☐ Fetch user statistics (skills count, progress, etc.)
- ☐ Display user welcome message

**Evening:**

- ☐ Add recent activity section
- ☐ Create quick action buttons
- ☐ Style and polish dashboard

**Deliverables:**

- ☒ Working dashboard with user stats
  - ☒ Visual progress indicators
  - ☒ Quick navigation to key features
- 

**Weekend (Nov 23-24) - Testing & Documentation 🕒 3-4 hours**

- ☐ Write API documentation for completed endpoints
- ☐ Test all authentication flows
- ☐ Fix any bugs discovered
- ☐ Update README with setup instructions
- ☐ Commit and push all code
- ☐ Plan Week 2 tasks

**Week 1 Milestone:** ☒ Authentication system complete

- ☒ User profile management working
  - ☒ Database fully operational
  - ☒ Basic frontend navigation ready
- 



## Week 2: Core Features

**Goal:** Implement CV upload, parsing, and skill management

**Day 6 (Nov 25) - CV Upload System 🕒 5-6 hours**

**Morning:**

- ☐ Backend: Configure Multer for file uploads
- ☐ Create uploads directory structure
- ☐ Build CV upload endpoint
- ☐ Add file validation (type, size)

**Afternoon:**

- ☐ Frontend: CV upload component
- ☐ Drag-and-drop file upload UI
- ☐ Upload progress indicator
- ☐ File preview functionality

**Evening:**

- ☐ Connect upload UI to backend

- ☐ Test with different PDF files
- ☐ Add error handling (file too large, wrong format)

**Deliverables:**

- ☒ Users can upload PDF CVs
  - ☒ Files stored securely
  - ☒ Upload feedback to users
- 

## Day 7 (Nov 26) - CV Parser Implementation 🕒 6-7 hours

**Morning:**

- ☐ Integrate `pdf-parse` library
- ☐ Create basic text extraction
- ☐ Build section identifier (skills, experience, education)

**Afternoon:**

- ☐ Create skill extraction logic
  - ☐ Pattern matching for common skills
  - ☐ Skills database lookup
- ☐ Parse work experience section
- ☐ Extract education information

**Evening:**

- ☐ Create CV parsing endpoint
- ☐ Test with various CV formats
- ☐ Store parsed data in database (JSONB)

**Deliverables:**

- ☒ CV text extraction working
  - ☒ Skills automatically detected
  - ☒ Experience and education parsed
- 

## Day 8 (Nov 27) - Skills Management 🕒 5-6 hours

**Morning:**

- ☐ Backend: Skills CRUD endpoints
  - ☐ GET all available skills
  - ☐ POST add skill to user
  - ☐ DELETE remove user skill
- ☐ Skill level tracking (beginner/intermediate/advanced)

**Afternoon:**

- ☐ Frontend: Skills management UI
- ☐ Display user skills with levels
- ☐ Add/remove skills interface

- ☐ Skills from CV vs. manually added

**Evening:**

- ☐ Integrate skills from parsed CV
- ☐ Auto-suggest skills while typing
- ☐ Visual skill level indicators

**Deliverables:**

- ☒ Complete skills management system
  - ☒ CV-extracted skills appear in profile
  - ☒ Manual skill addition/removal
- 

**Day 9 (Nov 28) - Career Paths Database 🕒 5-6 hours**

**Morning:**

- ☐ Create career paths seed data
  - ☐ Full-Stack Developer
  - ☐ Frontend Developer
  - ☐ Backend Developer
  - ☐ DevOps Engineer
  - ☐ Data Analyst
  - ☐ 5-10 more paths

**Afternoon:**

- ☐ Define skill requirements for each path
- ☐ Add salary ranges and market data
- ☐ Set up career-skills relationships

**Evening:**

- ☐ Create career paths API endpoints
- ☐ Build career path display component
- ☐ Test career data retrieval

**Deliverables:**

- ☒ 10+ career paths in database
  - ☒ Each with detailed requirements
  - ☒ API to fetch career information
- 

**Day 10 (Nov 29) - Skill Gap Analysis 🕒 5-6 hours**

**Morning:**

- ☐ Backend: Skill gap calculation logic
- ☐ Compare user skills vs career requirements
- ☐ Calculate match percentage

**Afternoon:**

- ☐ Create skill gap endpoint
- ☐ Frontend: Skill gap visualization
- ☐ Use Recharts for gap analysis chart

#### Evening:

- ☐ Display missing skills
- ☐ Show matched skills
- ☐ Visualize proficiency levels

#### Deliverables:

- ☒ Skill gap analysis functional
- ☒ Visual representation of gaps
- ☒ Match percentage calculated

### Weekend (Nov 30 - Dec 1) - Integration & Testing 🕒 3-4 hours

- ☐ Test entire CV upload → parse → skills flow
- ☐ Fix UI/UX issues
- ☐ Optimize database queries
- ☐ Update documentation
- ☐ Prepare for AI integration

#### Week 2 Milestone:

- ☒ CV upload and parsing complete
- ☒ Skills automatically extracted
- ☒ Skill gap analysis working
- ☒ Career paths database ready



## Week 3: AI Integration

**Goal:** Integrate OpenAI for career recommendations and chatbot

### Day 11 (Dec 2) - OpenAI Setup 🕒 4-5 hours

#### Morning:

- ☐ Get OpenAI API key
- ☐ Install OpenAI SDK
- ☐ Create OpenAI client wrapper
- ☐ Set up API key in environment

#### Afternoon:

- ☐ Build prompt templates
- ☐ Test basic OpenAI calls
- ☐ Implement error handling and retry logic

#### Evening:

- ☐ Create AI service architecture

- ☐ Plan prompt engineering strategy
- ☐ Document AI integration approach

**Deliverables:**

- ☒ OpenAI API connected
  - ☒ Basic AI calls working
  - ☒ Prompt templates ready
- 

**Day 12 (Dec 3) - Career Recommendation Engine 🕒 6-7 hours**

**Morning:**

- ☐ Build career advisor AI service
- ☐ Create recommendation prompt
- ☐ Include user context (skills, goals, experience)

**Afternoon:**

- ☐ Implement recommendation endpoint
- ☐ Parse AI response (JSON format)
- ☐ Validate and enrich recommendations

**Evening:**

- ☐ Frontend: Recommendations page
- ☐ Display career cards
- ☐ Show match scores and reasoning

**Deliverables:**

- ☒ AI-powered career recommendations
  - ☒ Top 3-5 paths suggested
  - ☒ Match scores and reasoning displayed
- 

**Day 13 (Dec 4) - Roadmap Generator 🕒 6-7 hours**

**Morning:**

- ☐ Design roadmap generation prompt
- ☐ Include career path requirements
- ☐ Structure output (phases, weeks, tasks)

**Afternoon:**

- ☐ Create roadmap generation endpoint
- ☐ Store roadmap in database
- ☐ Link to user and career path

**Evening:**

- ☐ Frontend: Roadmap viewer
- ☐ Display phases and weeks



- ☐ Show tasks with checkboxes
- ☐ Timeline visualization

**Deliverables:**

- ☒ AI generates personalized roadmaps
  - ☒ Roadmap stored in database
  - ☒ Visual roadmap interface
- 

**Day 14 (Dec 5) - AI Chatbot 🕒 6-7 hours**

**Morning:**

- ☐ Create chatbot service
- ☐ Build conversational prompt
- ☐ Include user context and history

**Afternoon:**

- ☐ Chat message endpoints
  - ☐ POST send message
  - ☐ GET chat history
- ☐ Store conversations in database

**Evening:**

- ☐ Frontend: Chat interface
- ☐ Message bubbles (user vs AI)
- ☐ Send message form
- ☐ Real-time updates

**Deliverables:**

- ☒ Working AI chatbot
  - ☒ Context-aware responses
  - ☒ Chat history saved
- 

**Day 15 (Dec 6) - Progress Tracking 🕒 5-6 hours**

**Morning:**

- ☐ Create progress tracking endpoints
- ☐ Log completed tasks
- ☐ Calculate progress percentage

**Afternoon:**

- ☐ Update roadmap task completion
- ☐ Track hours spent
- ☐ Record skill improvements

**Evening:**

- ☐ Frontend: Progress charts
- ☐ Weekly activity chart (Recharts)
- ☐ Task completion tracker
- ☐ Skill growth visualization

**Deliverables:**

- ☒ Progress tracking functional
  - ☒ Visual progress indicators
  - ☒ Task completion updates roadmap
- 

**Weekend (Dec 7-8) - AI Fine-tuning 🕒 4-5 hours**

- ☐ Test all AI features thoroughly
- ☐ Optimize prompts for better responses
- ☐ Add loading states for AI calls
- ☐ Implement AI response caching
- ☐ Fix any AI-related bugs

**Week 3 Milestone:**

- ☒ AI career recommendations working
  - ☒ Personalized roadmap generation
  - ☒ AI chatbot functional
  - ☒ Progress tracking implemented
- 



## Week 4: Polish & Deploy

**Goal:** Final touches, testing, and deployment

**Day 16 (Dec 9) - UI/UX Polish 🕒 5-6 hours**

**Morning:**

- ☐ Review entire app flow
- ☐ Improve responsive design
- ☐ Add loading skeletons
- ☐ Enhance error messages

**Afternoon:**

- ☐ Create consistent color scheme
- ☐ Add animations and transitions
- ☐ Improve button and form styling
- ☐ Mobile responsiveness check

**Evening:**

- ☐ Add success/error toast notifications
- ☐ Create empty states
- ☐ Improve navigation UX

**Deliverables:**

- ☒ Polished, professional UI
  - ☒ Smooth user experience
  - ☒ Mobile-friendly design
- 

## Day 17 (Dec 10) - Testing & Bug Fixes 🕒 5-6 hours

### Morning:

- ☐ User flow testing (register → login → CV → recommendations)
- ☐ Test edge cases
- ☐ Cross-browser testing

### Afternoon:

- ☐ Fix discovered bugs
- ☐ Optimize slow queries
- ☐ Add input validation everywhere

### Evening:

- ☐ Test error scenarios
- ☐ Security audit (SQL injection, XSS)
- ☐ Performance optimization

### Deliverables:

- ☒ Major bugs fixed
  - ☒ App stable and secure
  - ☒ Good performance
- 

## Day 18 (Dec 11) - Documentation 🕒 4-5 hours

### Morning:

- ☐ Complete README.md
- ☐ Add setup instructions
- ☐ Create screenshots for README

### Afternoon:

- ☐ Finalize API documentation
- ☐ Write code comments
- ☐ Create architecture diagrams

### Evening:

- ☐ Add LICENSE file
- ☐ Write CONTRIBUTING.md
- ☐ Create project demo video (optional)

### Deliverables:

- ☒ Professional documentation
- ☒ Clear setup guide

- ☒ Demo-ready presentation
- 

## Day 19 (Dec 12) - Environment Configuration 🕒 4-5 hours

### Morning:

- ☐ Create `.env.example` files
- ☐ Document all environment variables
- ☐ Set up production environment variables

### Afternoon:

- ☐ Configure backend for production
- ☐ Build frontend for production
- ☐ Test production builds locally

### Evening:

- ☐ Set up PostgreSQL on cloud (Neon/Supabase)
- ☐ Migrate database schema to cloud
- ☐ Test cloud database connection

### Deliverables:

- ☒ Production environment configured
  - ☒ Cloud database ready
  - ☒ Environment variables documented
- 

## Day 20 (Dec 13) - Deployment 🕒 6-7 hours

### Morning: Backend Deployment

- ☐ Create Render/Railway account
- ☐ Deploy backend to Render
- ☐ Configure environment variables
- ☐ Test backend endpoints

### Afternoon: Frontend Deployment

- ☐ Create Vercel/Netlify account
- ☐ Connect GitHub repository
- ☐ Configure build settings
- ☐ Deploy frontend

### Evening: Integration

- ☐ Update CORS settings
- ☐ Test frontend ↔ backend connection
- ☐ Fix deployment issues
- ☐ Set up custom domain (optional)

### Deliverables:

- ☒ Backend live and accessible
  - ☒ Frontend deployed
  - ☒ Full app working in production
- 

## Weekend (Dec 14-15) - Final Testing & Launch 🕒 3-4 hours

### Saturday:

- ☐ End-to-end production testing
- ☐ Fix any production bugs
- ☐ Optimize loading times
- ☐ Add analytics (Google Analytics/Plausible)

### Sunday:

- ☐ Create LinkedIn post about project
- ☐ Write detailed project description
- ☐ Update GitHub profile
- ☐ Share on Twitter/X
- ☐ Celebrate! 🎉

### Deliverables:

- ☒ Fully functional production app
  - ☒ Professional online presence
  - ☒ Ready for job applications
- 

## Day 21 (Dec 16) - Portfolio Integration 🕒 3-4 hours

### Morning:

- ☐ Create project showcase page
- ☐ Record demo video (3-5 minutes)
- ☐ Take screenshots of key features

### Afternoon:

- ☐ Update resume/CV with project
- ☐ Add to portfolio website
- ☐ Prepare technical interview talking points

### Evening:

- ☐ Write blog post about the project (optional)
- ☐ Share on GitHub discussions
- ☐ Reach out to recruiters

### Final Deliverables:

- ☒ Live, production-ready application
- ☒ Professional documentation
- ☒ Portfolio-ready demo
- ☒ Interview-ready talking points

---

## Daily Time Commitment

**Weekdays:** 5-7 hours/day

**Weekends:** 3-4 hours/day

**Total:** ~110-130 hours over 4 weeks

### Recommended Schedule:

- **Morning (2-3 hours):** Backend development
  - **Afternoon (2-3 hours):** Frontend development
  - **Evening (1-2 hours):** Testing, documentation, learning
- 

## Success Metrics

### Technical Milestones

- ☐ User authentication with JWT
- ☐ CV upload and parsing
- ☐ AI-powered career recommendations
- ☐ Personalized roadmap generation
- ☐ AI chatbot
- ☐ Progress tracking
- ☐ Deployed to production

### Quality Metrics

- ☐ All core features working
- ☐ Mobile responsive
- ☐ Fast loading (<3 seconds)
- ☐ No critical bugs
- ☐ Professional UI/UX

### Career Impact

- ☐ GitHub repository with 100+ commits
  - ☐ Professional README with screenshots
  - ☐ Live demo URL
  - ☐ Technical blog post (optional)
  - ☐ Added to resume/portfolio
- 

## Risk Mitigation

### Common Challenges & Solutions

Challenge	Solution
<b>OpenAI API costs too high</b>	Use gpt-4o-mini model, implement response caching, limit free tier usage

<b>CV parsing inaccurate</b>	Start with simple PDFs, improve iteratively, add manual correction option
<b>Database design changes</b>	Use migrations properly, version control schema changes
<b>Deployment issues</b>	Test locally with production builds first, read error logs carefully
<b>Time running out</b>	Focus on MVP, mark nice-to-have features as "future enhancements"
<b>Stuck on a feature</b>	Ask on Stack Overflow, check documentation, move to next task and return later

## MVP vs Future Features

### MVP (Must Have - 4 Weeks)

- ✓ User authentication
- ✓ Profile management
- ✓ CV upload and parsing
- ✓ Career recommendations
- ✓ Basic roadmap generation
- ✓ Simple chatbot
- ✓ Progress tracking

### Future Enhancements (Post-MVP)

- 📧 Email notifications
- 📧 Weekly AI coaching emails
- 🗣️ Community forum
- 📺 Course integration (Udemy API)
- 💼 Job board integration
- 👤 Mentor matching
- 📖 Interview preparation module
- 📄 Resume builder
- 📝 Skill assessments/quizzes

## Daily Checklist Template

### Before Starting:

- ☐ Review today's goals
- ☐ Check yesterday's incomplete tasks
- ☐ Set up dev environment

### During Development:

- ☐ Commit code frequently
- ☐ Test as you build
- ☐ Document complex logic

### Before Ending:

- ☐ Commit and push code

- ☐ Update progress tracker
  - ☐ Plan tomorrow's tasks
  - ☐ Note any blockers
- 



## Completion Checklist

By December 16, you should have:

### Code:

- ☐ GitHub repo with 100+ commits
- ☐ Clean, organized code
- ☐ All dependencies documented

### Documentation:

- ☐ Professional README
- ☐ API documentation
- ☐ Setup guide
- ☐ Architecture docs

### Deployment:

- ☐ Live backend URL
- ☐ Live frontend URL
- ☐ Working database
- ☐ SSL/HTTPS enabled

### Demo:

- ☐ Test account with sample data
- ☐ Demo video (3-5 min)
- ☐ Screenshots
- ☐ Feature list

### Career Materials:

- ☐ Updated resume
  - ☐ LinkedIn post
  - ☐ Portfolio entry
  - ☐ Interview prep notes
- 



## Next Steps After Completion

1. **Apply to Jobs** - Use this project in applications
  2. **Get Feedback** - Share with developers for code review
  3. **Iterate** - Implement one future feature per week
  4. **Blog** - Write technical articles about your journey
  5. **Open Source** - Accept contributions from others
-



**Remember:** This is an ambitious but achievable roadmap. Stay focused, commit daily, and you'll have a portfolio project that stands out!

**Good luck! You've got this! 🍊**

---

**Last Updated:** November 18, 2025