# 📅 1-Year Roadmap — Personal AI Friend Project

**Your Complete Learning & Building Plan | College Edition**

---

> **How This Works:**
>
> - **Mon–Fri:** Come to me every day. I give you that day's exact tasks.
> - **Saturday:** Revision of everything learned that week.
> - **Sunday:** Build a mini project using that week's skills.
> - **This document** = your big picture map. So you always know where you are and where you're going.

---

## 🗺️ THE BIG PICTURE

| Phase | Months | What You're Learning | What You're Building |
|---|---|---|---|
| 1 | 1–2 | Python + Git + APIs + Voice | Working Voice Chatbot with Memory |
| 2 | 3–4 | Backend + Databases + Auth | Chatbot with a real server and user login |
| 3 | 5–6 | AI Agents + RAG + LangGraph | Smart AI that knows YOU deeply |
| 4 | 7 | Computer Vision + OpenCV | Face scan → haircut recommendation |
| 5 | 8–9 | Web Scraping + Job Engine | Job readiness scorer + career agent |
| 6 | 10–11 | React Native + Mobile UI | Full mobile app on your phone |
| 7 | 12 | Deployment + Polish + Scale | Live, working product |

---

---

# 📆 MONTH 1 — Python + First AI Chatbot

**Theme:** Get comfortable with code. Build your first real AI feature. **End Goal:** A voice chatbot in your terminal that remembers you across sessions.

---

### Week 1 — Python Basics + Working Chatbot (Text)

**What you learn:**

- Python installation, VS Code setup, terminal commands
- Variables, data types, if/else, loops, functions
- Lists and dictionaries (your most used data structures)
- JSON files (how your AI remembers things)
- APIs — what they are, how to call them
- `.env` files and keeping secrets safe
- Git basics — init, add, commit, push

**Daily structure:** Come to me each morning. I give you that day's exact exercises, code to write, and concepts to understand.

**Saturday Revision Topics:**

- Variables, loops, functions — write 10 small programs from memory
- Review your chat.py line by line — explain every line out loud

**Sunday Mini Project:**

> **"Personal Profile App"** — Build a CLI app that asks the user 10 questions (name, age, goals, diet, sleep hours, etc.), saves the profile to a JSON file, and on next launch greets them by name and shows their profile. No AI yet — pure Python logic.

---

### Week 2 — Voice Input/Output (Whisper + ElevenLabs)

**What you learn:**

- Recording audio from microphone in Python (`sounddevice`, `pyaudio`)
- OpenAI Whisper API — converting your voice to text
- ElevenLabs API — converting text to a realistic voice
- Playing audio files in Python (`pygame`, `playsound`)
- Combining: voice in → Claude brain → voice out

**Saturday Revision Topics:**

- Rebuild the voice pipeline from scratch without looking at previous code
- Explain the full voice flow: mic → Whisper → Claude → ElevenLabs → speaker

**Sunday Mini Project:**

> **"Voice Note Taker"** — Speak a note into your mic. App transcribes it using Whisper, saves it to a `.txt` file with a timestamp, and reads it back to you using ElevenLabs. Feels like a real product.

---

### Week 3 — Persistent Memory + Personality

**What you learn:**

- Saving and loading conversation history from JSON
- System prompts — giving your AI a real personality
- Structuring the conversation history array correctly
- Error handling with try/except (what happens when API fails)
- Basic Python classes (start thinking in objects)

**Saturday Revision Topics:**

- Rebuild the memory system from scratch
- Experiment with 5 different system prompts — notice how personality changes

**Sunday Mini Project:**

> **"Fitness Check-in Bot"** — A voice bot that every day asks: how many hours did you sleep, did you work out, what did you eat. Saves answers to JSON. On next session, it references yesterday's answers and gives feedback. Your first real useful daily tool.

---

### Week 4 — Polish + GitHub + Month Review

**What you learn:**

- Making your chatbot feel more natural (conversation flow improvements)
- Writing a proper README on GitHub
- Python virtual environments deeply (why they matter)
- Code organization — separating code into multiple files
- `requirements.txt` — how to share your project so others can run it

**Saturday Revision:**

- Full month review. Rebuild the entire chatbot from a blank file. No copy-paste.

**Sunday Mini Project (Month 1 Capstone):**

> **"MAX v1.0 — Your AI Voice Friend"** — Full voice chatbot with persistent memory, a real personality via system prompt, clean error handling, organized code across multiple files, and pushed to GitHub with a proper README. This is your first real portfolio project.

---

---

# 📅 MONTH 2 — Deepen Python + Smarter AI Memory

**Theme:** Stop writing beginner code. Start writing organized, real-world code. **End Goal:** MAX remembers specific facts about you (not just conversation), has multiple modes (fitness mode, career mode), and handles edge cases gracefully.

---

### Week 5 — Intermediate Python

**What you learn:**

- Object-Oriented Programming (OOP) — classes, objects, methods, inheritance
- Why OOP matters: your chatbot, your user profile, your memory — all become objects
- File management — organizing a real project structure
- List comprehensions, lambda functions, map/filter
- Python modules — importing your own code across files

**Saturday Revision:**

- Rewrite your chatbot using classes. `ChatBot` class, `Memory` class, `VoiceEngine` class.

**Sunday Mini Project:**

> **"Workout Logger OOP"** — A class-based workout logger. `Workout` class with attributes (date, exercises, sets, reps). `WorkoutLog` class that saves/loads from JSON and prints weekly summaries. Clean, object-oriented code.

---

### Week 6 — Structured Memory (User Profile + Long-Term Facts)

**What you learn:**

- The difference between conversation history and long-term facts
- Building a user profile system (name, goals, preferences, physical stats)
- Injecting user profile into the system prompt dynamically
- Updating facts when the user mentions changes ("I changed my goal to lose weight")
- LLM-based fact extraction — asking Claude to extract facts from conversation

**Saturday Revision:**

- How does your memory system work? Draw the full data flow on paper.

**Sunday Mini Project:**

> **"Smart Profile Updater"** — After every conversation, a second Claude call analyzes the conversation and extracts any new facts about the user (new goals, habits, preferences). Updates the profile JSON automatically. MAX now gets smarter the more you talk to him.

---

## Week 7 — Multiple Modules (Fitness + Diet Mode)

**What you learn:**

- Structuring a multi-module app (separate files for fitness, diet, career, etc.)
- Conditional routing — detecting what topic the user is talking about
- Writing specialized system prompts per domain
- Logging and tracking data over time (streaks, progress)

**Saturday Revision:**

- How do you detect intent from user speech? What approaches work?

**Sunday Mini Project:**

> **"Fitness + Diet Advisor"** — Voice-based. User can say "fitness mode" or "diet mode." Each mode has a specialized AI personality. Fitness mode tracks workouts. Diet mode logs meals and gives macro estimates. Data saved separately per module.

---

## Week 8 — Error Handling, Logging, Month Review

**What you learn:**

- Robust error handling (API failures, missing files, bad input)
- Python `logging` module — writing logs to a file instead of just printing
- Code review mindset — reading your own old code and improving it
- Writing clean, readable code (comments, naming conventions, structure)

**Saturday Revision:**

- Full month review.

**Sunday Mini Project (Month 2 Capstone):**

> **"MAX v2.0"** — Upgraded chatbot with OOP architecture, long-term user profile, fitness and diet modules, automatic fact extraction after each session, proper logging, and clean project structure. This is a genuinely impressive project at this stage.

---

---

# 📅 MONTH 3 — Backend Fundamentals (FastAPI)

**Theme:** Give your AI a real server brain. Stop running everything locally. **End Goal:** Your chatbot runs as an API server. Any device can connect to it.

---

### Week 9 — What is a Backend + FastAPI Basics

**What you learn:**

- What a backend server is and why you need one
- HTTP — GET, POST, PUT, DELETE (how the internet actually works)
- FastAPI — Python's best framework for building APIs
- Endpoints, request bodies, response models
- Pydantic — data validation in Python
- Running a local server and testing it with Postman or curl

**Saturday Revision:**

- Build 5 different API endpoints from scratch

**Sunday Mini Project:**

> **"Workout API"** — A FastAPI server with endpoints to: add a workout (POST), get all workouts (GET), get a specific workout by date (GET), delete a workout (DELETE). Test everything in Postman.

---

### Week 10 — Databases (PostgreSQL)

**What you learn:**

- Why JSON files break at scale (and when to move to a real DB)
- PostgreSQL — installation, basic SQL queries
- SELECT, INSERT, UPDATE, DELETE
- Connecting FastAPI to PostgreSQL using `SQLAlchemy` or `asyncpg`
- Database migrations — changing your DB structure safely

**Saturday Revision:**

- Write 20 SQL queries from memory (SELECT with filters, JOINs, aggregates)

**Sunday Mini Project:**

> **"User Profile DB"** — Move your user profile from JSON to PostgreSQL. FastAPI endpoints to create profile, update profile, retrieve profile. All data stored in a real database now.

---

## Week 11 — Authentication (User Login System)

**What you learn:**

- JWT tokens — how login sessions work
- Hashing passwords with `bcrypt`
- OAuth2 with FastAPI
- Protecting routes — only logged-in users can access certain endpoints
- Why authentication matters (your app needs this before anyone else can use it)

**Saturday Revision:**

- Explain the full login flow: user signs up → password hashed → stored in DB → user logs in → JWT issued → user sends JWT with each request → server verifies → access granted

**Sunday Mini Project:**

> **"Secure MAX API"** — Your chatbot API now has login. Users register, log in, get a JWT token, and send that token with every message. Each user has their own separate memory and profile stored in the database.

---

## Week 12 — Connecting Frontend to Backend + Month Review

**What you learn:**

- CORS — why browsers block API calls and how to fix it
- Environment variables on a server
- Structuring a real FastAPI project (routers, models, schemas, services)
- Basic API documentation with FastAPI's built-in Swagger UI

**Saturday Revision:**

- Full month review.

**Sunday Mini Project (Month 3 Capstone):**

> **"MAX Backend v1.0"** — Full FastAPI server with PostgreSQL database, user authentication with JWT, chat endpoint that stores full conversation history per user in the DB, profile endpoint, and workout logging endpoint. Your chatbot now has a real production-grade backend.

---

---

# 📅 MONTH 4 — Advanced Backend + Redis + Async

**Theme:** Make your backend fast, real-time, and production-ready. **End Goal:** Notifications, real-time task alerts, and a blazing fast server.

---

### Week 13 — Redis + Caching

**What you learn:**

- What Redis is (in-memory data store — blazing fast)
- Caching API responses (so repeated requests don't call Claude every time)
- Redis for storing sessions
- When to use Redis vs PostgreSQL

**Sunday Mini Project:**

> **"Smart Caching"** — If the user asks the same fitness question within 1 hour, return the cached answer instead of calling Claude again. Saves API costs. Learns good engineering habits.

---

### Week 14 — Background Tasks + Scheduling

**What you learn:**

- FastAPI background tasks
- APScheduler — running code on a schedule in Python
- Celery + Redis — for heavy background work
- Push notifications setup (Firebase FCM)

**Sunday Mini Project:**

> **"Daily Routine Alerter"** — User sets a daily schedule (wake up 6am, gym 7am, study 9am). Server sends them a voice notification at the right time. First version of your routine tracker feature.

---

### Week 15 — MongoDB + Flexible Data Storage

**What you learn:**

- When to use MongoDB vs PostgreSQL
- MongoDB basics — documents, collections, queries
- Using `motor` (async MongoDB driver) with FastAPI

- Storing conversation history in MongoDB (perfect fit — flexible structure)

**Sunday Mini Project:**

> **"Conversation Archive"** — Move all conversation history to MongoDB. Build an endpoint to search past conversations by keyword. "Show me everything I've said about my diet in the last 30 days."

---

### Week 16 — Async Python + Month Review

**What you learn:**

- `async` and `await` in Python (non-blocking code)
- Why async matters for your voice app (don't freeze while waiting for API)
- Async FastAPI endpoints
- Load testing basics — making sure your server doesn't crash

**Sunday Mini Project (Month 4 Capstone):**

> **"MAX Backend v2.0"** — Full async FastAPI server, PostgreSQL for user data, MongoDB for conversations, Redis for caching and sessions, background task scheduler for daily alerts, and Firebase push notifications. This is a startup-grade backend.

---

---

## 📆 MONTH 5 — AI Agents + LangGraph

**Theme:** Your AI stops just chatting. It starts DOING things. **End Goal:** MAX can take actions — set reminders, look up information, analyze your data.

---

### Week 17 — LangChain Fundamentals

**What you learn:**

- What LangChain is and why it exists
- Chains — linking multiple AI calls together
- Tools — giving your AI the ability to call functions
- LLM-based decision making: "Should I call the fitness tool or the diet tool?"

**Sunday Mini Project:**

> **"Tool-Using MAX"** — MAX can now call a `get_weather` tool, a `calculate_macros` tool, and a `get_workout_plan` tool. The AI decides which tool to call based on what you ask.

---

## Week 18 — RAG (Teach MAX About YOU)

**What you learn:**

- What RAG (Retrieval Augmented Generation) is
- Vector databases — how AI finds relevant information
- Embeddings — turning text into mathematical vectors
- ChromaDB or Pinecone — storing and searching embeddings
- Building a personal knowledge base for your AI

**Saturday Revision:**

- Explain RAG like you're explaining it to a 10-year-old. If you can, you understand it.

**Sunday Mini Project:**

> **"MAX Knows Your History"** — Upload 3 months of your fitness logs, diet notes, and goals into a vector database. MAX can now answer: "How has my sleep pattern affected my workout performance over the last month?" without you telling him anything — he retrieves it himself.

---

## Week 19 — LangGraph (Multi-Step AI Agents)

**What you learn:**

- LangGraph — building AI agents that can plan and execute multi-step tasks
- State machines for AI workflows
- Agents that can search, reason, and act in loops until they finish a task
- Error recovery — what happens when an agent gets stuck

**Sunday Mini Project:**

> **"Morning Brief Agent"** — Every morning, MAX runs an agent that: checks your schedule for the day, checks your pending tasks, reviews your fitness goals, and delivers a personalized 2-minute voice briefing. Fully automated. No prompting needed.

---

## Week 20 — Finance Module + Month Review

**What you learn:**

- Working with financial data in Python
- Manual transaction logging (for now — no bank API yet)
- Basic budgeting logic and savings analysis

- LLM-based spending pattern analysis

**Sunday Mini Project (Month 5 Capstone):**

> **"MAX Agent v1.0"** — MAX is now a real agent. He has tools, a personal knowledge base with your history (RAG), can run multi-step tasks autonomously, delivers daily morning briefs, and has a basic finance tracker. This is the most impressive version yet.

---

---

## 📅 MONTH 6 — Advanced Agents + Job Readiness Engine

**Theme:** Build your killer feature — the thing nobody else has. **End Goal:** MAX knows your skills, tracks your growth, and only sends you job alerts when you're ready.

---

### Week 21 — Skill Graph + Resume Parser

**What you learn:**

- NLP basics — extracting keywords and skills from text
- Parsing your own resume (PDF parsing with `pypdf2`)
- Building a skills inventory from your resume
- Comparing skills to job descriptions programmatically

**Sunday Mini Project:**

> **"Resume Skill Extractor"** — Upload your resume. The system extracts all your skills, categorizes them (languages, frameworks, soft skills), and stores them in your profile. Foundation of the job engine.

---

### Week 22 — GitHub + LeetCode Integration

**What you learn:**

- GitHub API — fetching your repos, commit history, languages used
- Analyzing your GitHub to infer real skill level (not just claimed skills)
- LeetCode API / scraping — tracking your problem-solving progress
- Building a composite skill score: resume + GitHub + LeetCode

**Sunday Mini Project:**

> **"Real Skill Score"** — Connect GitHub. System scans your repos, identifies languages and frameworks actually used (not just listed on resume), and generates a skill confidence score. 0–100 per skill. Honest,

> data-driven.

---

### Week 23 — Job Matching Engine

**What you learn:**

- Web scraping with `BeautifulSoup` and `Playwright`
- Scraping Unstop, LinkedIn Jobs, Internshala
- Job description parsing — extracting required skills
- Building a match score: your skills vs job requirements
- Threshold alerting — only notify when match score > 75%

**Sunday Mini Project:**

> **"Job Readiness Scorer"** — Paste a job description. System scores your readiness 0–100, tells you exactly which skills you have, which you're missing, and gives you a learning plan to close the gap. This is a standalone product.

---

### Week 24 — Automated Job Agent + Month Review

**What you learn:**

- Scheduled agent that runs daily job searches
- Filtering and ranking results by readiness score
- Sending you only the jobs you're actually ready for
- Email/notification integration

**Sunday Mini Project (Month 6 Capstone):**

> **"Career Agent v1.0"** — Fully automated job agent. Scans job boards daily. Scores every opportunity against your real skills. Sends you a voice notification only when you cross the readiness threshold. Includes what to study to get the next job faster. This feature is genuinely hireable work.

---

---

## 📅 MONTH 7 — Computer Vision (Face + Grooming)

**Theme:** Give MAX eyes. **End Goal:** Point your camera at yourself and get personalized grooming advice.

---

## Week 25 — OpenCV Fundamentals

**What you learn:**

- OpenCV — the world's most popular computer vision library
- Reading images and video streams in Python
- Basic image processing (resize, crop, color conversion, filters)
- Drawing on images (bounding boxes, landmarks)

**Sunday Mini Project:**

> **"Face Detector"** — Open your webcam. App draws a bounding box around your face in real time. Counts how many faces are in frame. Sounds simple but teaches you the entire OpenCV pipeline.

---

## Week 26 — MediaPipe + Face Shape Detection

**What you learn:**

- MediaPipe — Google's framework for face landmarks, pose, hands
- 468 face landmark points — what they are and how to use them
- Calculating face shape from landmark geometry (oval, round, square, oblong, heart)
- Mapping face shapes to recommended hairstyles

**Sunday Mini Project:**

> **"Face Shape Analyzer"** — Upload a photo or use webcam. App detects your face shape (oval, round, square, etc.) and MAX tells you via voice which hairstyles suit you and why. Add beard recommendations too. Add glasses frame recommendations.

---

## Week 27 — Grooming Full Feature + Outfit Advice

**What you learn:**

- Skin tone detection from face region
- Color theory — which clothing colors suit which skin tones
- Integrating vision output with LLM for detailed personalized advice
- Handling different lighting conditions in images

**Sunday Mini Project:**

> **"Style Advisor"** — Upload a photo. MAX analyzes face shape, skin tone, and (optionally) what you're wearing. Gives you personalized advice on haircut, beard style, clothing colors, and accessories. Feels like a real product.

---

### Week 28 — Month Review + Vision Integration

**What you learn:**

- Connecting the vision module to your FastAPI backend
- Sending images from mobile to backend for analysis
- Caching vision results (face shape doesn't change daily)

**Sunday Mini Project (Month 7 Capstone):**

> **"MAX Vision v1.0"** — Complete grooming module integrated into MAX. User takes a photo via app (or uploads). MAX analyzes face shape, skin tone, suggests hairstyles with visual references (image search via API), and gives a full grooming routine recommendation via voice.

---

---

## 📅 MONTH 8–9 — React Native (Mobile App)

**Theme:** Put MAX in your pocket. **End Goal:** A real mobile app on your phone with voice, camera, dashboard, and all modules.

---

### Month 8 — React Native Fundamentals

#### Week 29–30 — JavaScript + React Basics

**What you learn:**

- JavaScript fundamentals — variables, functions, arrays, objects, async/await
- React basics — components, props, state, hooks (`useState`, `useEffect`)
- Why JavaScript for mobile: one language for everything frontend

**Sunday Projects:**

- Week 29: A React to-do list app in browser (foundational)
- Week 30: A React fitness dashboard showing workout data from your backend API

#### Week 31–32 — React Native + Expo

**What you learn:**

- React Native vs React — what changes, what stays the same
- Expo — the fastest way to build React Native apps

- Core components: `View`, `Text`, `TextInput`, `TouchableOpacity`, `FlatList`, `ScrollView`
- Styling in React Native (Flexbox)
- Navigation between screens (`react-navigation`)

**Sunday Projects:**

- Week 31: A multi-screen app with navigation (Home, Profile, Settings)
- Week 32: Connect one screen to your real FastAPI backend — show user profile data

---

## Month 9 — Full App Build

### Week 33 — Voice Interface Screen

**What you learn:**

- Accessing microphone in React Native (`expo-av`)
- Recording audio and sending to your backend (which runs Whisper)
- Playing audio responses from ElevenLabs
- Building the voice chat UI — a clean, minimal conversation screen

**Sunday Mini Project:**

> **"Voice Chat Screen"** — The main screen of MAX. Press button, speak, release. MAX thinks, responds with voice. Conversation displayed on screen. This is your hero feature in mobile form.

### Week 34 — Camera + Vision Screen

**What you learn:**

- Accessing camera in React Native (`expo-camera`)
- Capturing photos and sending to backend for analysis
- Displaying vision results beautifully in the app

**Sunday Mini Project:**

> **"Grooming Scanner Screen"** — Open camera, take selfie, send to backend, display face shape + recommendations on screen with MAX's voice narration.

### Week 35 — Dashboard + All Modules UI

**What you learn:**

- Building a dashboard UI (fitness stats, finance summary, career progress)
- Charts in React Native (`victory-native` or `react-native-chart-kit`)
- Notifications in React Native (connecting Firebase)

- Local storage for offline capability (`AsyncStorage`)

**Sunday Mini Project:**

> **"Full Dashboard"** — Home screen showing: today's tasks, fitness streak, current skill score, finance summary, and next recommended job opportunity. Everything in one view.

#### Week 36 — Polish + Month Review

**Sunday Mini Project (Month 9 Capstone):**

> **"MAX App v1.0"** — A real mobile app. Voice chat screen, grooming scanner, fitness tracker, career dashboard, routine alerts. Works on your actual phone. This is your portfolio centerpiece.

---

---

## 📅 MONTH 10 — Integration + Advanced Features

**Theme:** Make everything talk to each other. Fill the gaps. **End Goal:** All 6 modules connected. The app feels like one cohesive product.

---

### Week 37 — LinkedIn + GitHub Profile Integration

**What you learn:**

- LinkedIn API (limited but useful)
- GitHub OAuth login — user logs in with GitHub
- Pulling real data automatically instead of manual input
- Keeping skill scores updated as you push new code

**Sunday Mini Project:**

> **"Auto Skill Sync"** — Connect GitHub. Every time you push code, MAX automatically updates your skill scores. Career section always reflects your real, current skill level.

---

### Week 38 — Finance Deep Dive

**What you learn:**

- Manual expense logging with smart categorization (LLM categorizes "Zomato" as Food)
- Budget setting and alerts
- Savings goal tracking
- Monthly financial report generation

**Sunday Mini Project:**

> **"Smart Finance Tracker"** — Log an expense by voice: "I spent 500 rupees on food at Zomato." MAX categorizes it, adds to monthly total, and warns you if you're approaching your food budget.

---

### Week 39 — Routine Tracker Deep Dive

**What you learn:**

- Habit tracking data models
- Streak calculation
- Smart scheduling — learning your patterns and suggesting optimal times
- Daily/weekly/monthly review generation

**Sunday Mini Project:**

> **"Habit Engine"** — Track 5 daily habits. See streaks. Get a Monday morning voice briefing: "Last week you hit 4/7 workout days, 7/7 sleep goals, and 5/7 study sessions. Here's your focus for this week."

---

### Week 40 — Advanced Personalization

**What you learn:**

- Learning user patterns over time (sleep times, workout frequency, spending peaks)
- Proactive notifications — MAX notifies you before you need to ask
- User feedback loop — thumbs up/down on advice to improve recommendations

**Sunday Mini Project (Month 10 Capstone):**

> **"MAX v3.0 — Full Integration"** — All 6 modules fully connected. Proactive alerts. Personalized recommendations based on 10 months of architecture. The app feels like it genuinely knows you.

---

---

## 📅 MONTH 11 — Testing + Optimization

**Theme:** Make it not break. Make it fast. Make it secure. **End Goal:** An app you can confidently show to anyone and demo live.

---

### Week 41 — Testing

**What you learn:**

- Unit testing in Python with `pytest`
- Testing API endpoints with `httpx`
- Why testing is not optional — it catches bugs before users do
- Test-driven development (TDD) mindset

**Sunday Mini Project:**

> Write a complete test suite for your job readiness engine. Every function tested. Every edge case covered. This is what professional code looks like.

---

### Week 42 — Performance Optimization

**What you learn:**

- Profiling your Python code (finding slow parts)
- Database query optimization (indexes, query plans)
- API response time optimization
- Reducing voice response latency (the gap between speaking and hearing MAX respond)

**Sunday Mini Project:**

> Benchmark your voice chatbot. Measure time from speaking to response. Optimize until it's under 3 seconds end-to-end.

---

### Week 43 — Security Audit

**What you learn:**

- Common security vulnerabilities (SQL injection, XSS, insecure tokens)
- Rate limiting your API
- HTTPS setup
- Secure API key management
- Input validation everywhere

---

### Week 44 — Beta Testing + Feedback

**What you learn:**

- Getting 3–5 real users to test your app (friends/classmates)
- Collecting and prioritizing feedback

- Fixing the most impactful bugs and UX issues

**Sunday Mini Project (Month 11 Capstone):**

> **"Beta Demo"** — Demo MAX to 5 real people. Record their reactions. Fix the top 5 issues they find. Getting real user feedback is a skill in itself.

---

---

## 📅 MONTH 12 — Deployment + Launch

**Theme:** Make it real. Put it on the internet. Show the world. **End Goal:** MAX is live. Anyone can sign up and use it.

---

### Week 45 — Docker

**What you learn:**

- What Docker is (packaging your app so it runs anywhere)
- Writing a `Dockerfile`
- `docker-compose` — running your backend + database + Redis together
- Why this matters: "it works on my machine" never happens again

**Sunday Mini Project:**

> Dockerize your entire backend. Run everything with one command: `docker-compose up`.

---

### Week 46 — Cloud Deployment

**What you learn:**

- Railway.app or Render — simplest cloud deployment for beginners
- Environment variables in production
- Domain names and HTTPS
- Monitoring and uptime alerts

**Sunday Mini Project:**

> Deploy MAX backend to the cloud. Get a live URL. Call your API from your phone using the real URL, not localhost.

---

### Week 47 — App Store Prep (Optional but powerful)

**What you learn:**

- Expo EAS Build — building a real APK/IPA from your React Native app
- App Store Connect and Google Play Console basics
- App icon, splash screen, and metadata

---

### Week 48 — Final Polish + Launch + Portfolio

**What you learn:**

- Writing a compelling project description for LinkedIn and GitHub
- Recording a demo video of MAX in action
- How to talk about this project in an interview

**Sunday Final Project (Year 1 Capstone):**

> **"MAX v1.0 — Public Launch"**
>
> - Live backend on the cloud
> - Mobile app on your phone
> - GitHub repo with clean README and architecture diagram
> - Demo video on LinkedIn
> - Case study post explaining what you built, what you learned, what problems you solved
>
> This project, documented properly, puts you ahead of 95% of CS students in India for internships and jobs.

---

---

## 📊 SKILLS YOU WILL HAVE AFTER 1 YEAR

| Skill | Level |
|---|---|
| Python | Advanced |
| FastAPI + Backend Architecture | Intermediate-Advanced |
| SQL + NoSQL Databases | Intermediate |
| LLM APIs + Prompt Engineering | Advanced |
| AI Agents (LangGraph) | Intermediate |

| Skill | Level |
|---|---|
| RAG + Vector Databases | Intermediate |
| Computer Vision (OpenCV + MediaPipe) | Intermediate |
| React Native Mobile Development | Intermediate |
| Git + GitHub | Advanced |
| Docker + Cloud Deployment | Intermediate |
| System Design Thinking | Intermediate |
| Voice AI (Whisper + ElevenLabs) | Advanced |

---

## 💡 GROUND RULES FOR THE YEAR

**1. Every day you come, I give you that day's task.** Tell me which day of which week you're on and I'll give you exact exercises, code, and explanations.

**2. Never skip the Sunday mini project.** The concepts will not stick without building something with them. Theory alone is worthless.

**3. Push to GitHub every single session.** Your commit history is your proof of work. By month 6, it'll look incredible.

**4. Struggle is the process.** When something doesn't work for 2 hours, that's not failure — that's the actual learning happening. Come to me when you're stuck, not when you've given up.

**5. Build for yourself first.** Use MAX every single day. The best product feedback comes from being your own most honest user.

**6. Document as you go.** Write a LinkedIn post every month about what you built. Build your audience while you build your product.

---

> **You now have the map. Come back tomorrow and we start Day 1 of Week 1. The only thing that separates you from finishing this is showing up every day.**