YOUR PERSONALIZED PATH · UPDATED FOR 2026

AI Engineering Roadmap

From MERN + TypeScript to AI Agents & AI Engineering — paced alongside your BS Data Science degree

You've completed MERN, TypeScript, and Next.js, and you're now learning Python. You're also newly admitted to the BS Data Science program at Virtual University of Pakistan — and since the first two semesters carry no data-science-specific coursework, this roadmap is sequenced to run in that window. Finish this while your semesters are still general-education-heavy, so that by the time your program's data science courses begin, you're walking in as a working AI engineer, not a beginner.

13	2	1
Phases	Done (Next.js)	Capstone

Built from roadmap.sh's AI Engineer and AI Agents roadmaps, reordered and paced for your MERN + TypeScript background — and revised for 2026 tooling: LangGraph over legacy LangChain, MCP given real weight, and LlamaIndex + LangGraph used together rather than as alternatives.

CONTENTS

Roadmap at a Glance

0	Where You Stand ✎ revised	Starting point — updated
1	Next.js ✎ revised	2–3 weeks — COMPLETED
2	Python Fundamentals	2–3 weeks — in progress
3	FastAPI	2–3 weeks · run alongside Phase 4
4	LLM Fundamentals	run alongside Phase 3
5	Prompt Engineering	~1 week
6	Embeddings & Vector Databases	1–2 weeks
7	RAG (Retrieval-Augmented Generation) ✎ revised	2 weeks — updated
8	AI Agents 101	1–2 weeks
9	Agent Memory	~1 week
10	Agent Architectures ✎ revised	1–2 weeks — updated
11	Agent Frameworks ✎ revised	1–2 weeks — updated
12	Evaluation, Observability & Security	1–2 weeks
★	Capstone Project ✎ revised	— updated

Where a phase says “overlaps,” start it before finishing the previous one. Items marked “revised” have 2026 tooling updates called out inline.

THE ROADMAP

Phase by Phase

0 Where You Stand

Starting point — updated

TOPICS

- MERN stack (MongoDB, Express, React, Node.js)
- TypeScript
- **Next.js — complete**
- Newly admitted: BS Data Science, Virtual University of Pakistan (first two semesters have no data-science-specific courses — use this window for this roadmap)

2026 UPDATE

> Original plan assumed you'd start Next.js next. You've since finished it and moved on to Python — the roadmap below reflects that. Your BSDS timeline is also now the backdrop for pacing: aim to clear most of this roadmap before your program's data-science-specific courses begin, so you enter them as a working AI engineer rather than starting cold.

---

1 Next.js

2–3 weeks

✓ COMPLETED

TOPICS

- App router & file-based routing
- SSR, SSG, and ISR
- API routes & server actions
- Deployment on Vercel

PROJECT

> Rebuild one existing MERN project in Next.js.

---

2 Python Fundamentals

2–3 weeks · overlaps Phase 1

■ IN PROGRESS NOW

TOPICS

- Syntax & data structures

- OOP in Python
- Async / await
- venv & pip

PROJECT

> A small CLI script — focus on fluency, not scope.

---

3 FastAPI

2–3 weeks · run alongside Phase 4

TOPICS

- Routes & Pydantic schemas
- Async endpoints
- Database layer (SQLAlchemy / Postgres)
- Authentication

PROJECT

> Clone one Express API you've already built, in FastAPI.

---

4 LLM Fundamentals

run alongside Phase 3

TOPICS

- Transformers, tokenization, context windows, token pricing
- Generation controls: temperature, top-p, frequency / presence penalty
- Open vs. closed weight models · reasoning vs. standard models
- OpenAI / Anthropic API basics: chat completions, playground, fine-tuning basics

PROJECT

> A Python CLI chatbot hitting an LLM API directly.

---

5 Prompt Engineering

~1 week

TOPICS

- Specificity, context, and relevant technical terms
- Using examples & iterating on prompts
- ReAct prompting

PROJECT

> A small prompt-testing harness.

---

6 Embeddings & Vector Databases

1–2 weeks

TOPICS

- Semantic search, recommendations, anomaly detection
- OpenAI embeddings + one open-source option (Sentence Transformers)
- Pick one vector DB to start — Chroma is still the easiest on-ramp

PROJECT

Semantic search over your own notes or docs.

7 RAG (Retrieval-Augmented Generation)

2 weeks — updated

TOPICS

- Chunking → embedding → vector DB → retrieval → generation
- Direct SDK implementation first, then a retrieval framework

2026 UPDATE

> Learn this with **LlamaIndex** rather than treating LangChain/LlamaIndex as interchangeable. By 2026 they've specialized: LlamaIndex is the stronger pick for document-centric RAG and retrieval quality (chunking strategies, query decomposition, response synthesis). Save LangChain for the orchestration layer in Phase 11.

PROJECT

> RAG chatbot over a PDF — FastAPI backend, Next.js frontend. This is where everything connects.

---

8 AI Agents 101

1–2 weeks

TOPICS

- Agent loop: perceive → reason / plan → act / invoke tool → observe / reflect
- Tool definition: schema & error handling
- Start with one tool: web search, calculator, or a DB query

PROJECT

> A single-tool agent.

---

9 Agent Memory

~1 week

TOPICS

- Short-term vs. long-term memory
- Episodic vs. semantic memory
- Summarization / compression · forgetting & aging strategies

---

10 Agent Architectures

1–2 weeks — updated

TOPICS

- ReAct · Chain of Thought · RAG-agent
- Planner-Executor · DAG agents · Tree-of-Thought
- Model Context Protocol (MCP): hosts, clients, servers

2026 UPDATE

> MCP has grown from a minor topic into a real differentiator — it's becoming the enterprise standard for connecting AI models to external tools and data, and engineers who can actually *build* MCP servers (not just call them) are still a small minority. Don't just read about MCP here — ship a minimal server.

PROJECT

> Multi-tool ReAct agent with memory — built manually first (no framework) so you understand the loop. Then build one small custom MCP server exposing a tool of your own.

---

11 Agent Frameworks

1–2 weeks — updated

TOPICS

- LangGraph (agent orchestration, state, persistence)
- LlamaIndex Workflows (retrieval-heavy pipelines)
- CrewAI · AutoGen (multi-agent collaboration, for exposure)
- Anthropic tool use

2026 UPDATE

> Plain LangChain (LLMChain, initialize_agent) is now legacy — its center of gravity moved to **LangGraph**, which is the part actually worth learning: durable state, persistence, first-class human-in-the-loop. The 2026 default pattern is LlamaIndex for retrieval + LangGraph for orchestration, used together rather than picked as alternatives.

PROJECT

> Rebuild your Phase 10 agent using LangGraph, wired to your Phase 7 LlamaIndex retrieval layer — compare against your manual version and you'll understand why the framework exists.

---

12 Evaluation, Observability & Security

1–2 weeks

TOPICS

- Unit & integration testing for tools
- Human-in-the-loop evaluation: LangSmith, Ragas, DeepEval
- Prompt injection defense, tool sandboxing, PII redaction, red-teaming

---

★ Capstone Project

— updated

TOPICS

- Next.js frontend + FastAPI backend
- LlamaIndex retrieval layer over real documents
- LangGraph agent orchestration, multi-tool, with memory
- At least one custom MCP server exposed as a tool
- Deployed and observable in production

PROJECT

> Full-stack AI agent product — this is your portfolio piece.

The biggest risk isn't the order — it's spending too long “preparing” in any one phase before building something. Ship something small at the end of every phase, even if it's ugly. And keep this roadmap moving now, while your BSDS semesters are still general-ed — the payoff is starting your program's actual data science courses already employable.