

Supernova 6-Month Tech Syllabus

1. FULL STACK + AI INTEGRATION

- *Goal:* Build powerful full stack web apps that are smart, responsive, and AI-driven.
 - ❖ *Outcome:* Placement-ready portfolio + Real projects + Deep understanding.
-

🚧 PHASE 0: Prep Setup

- VS Code + Extensions
- GitHub Account & Daily Push Ritual
- Codepen + Postman Setup
- Python + Node.js installed
- Folder Structure Ready (/frontend, /backend, /ml)

◆ PHASE 1: Web Development Foundation (HTML + CSS + JS)

📅 **Duration:** Week 1–2

📖 **Topics:**

- HTML5 (semantic tags, forms, structure)
- CSS3 (Flexbox, Grid, Media Queries)
- JavaScript (variables, functions, arrays, loops)
- DOM Manipulation + Events

💡 **Mini Projects:**

- Responsive Portfolio
- Animated Button Set
- To-Do List (JS DOM based)

🌟 PHASE 2: React.js + Tailwind CSS+TypeScript

📅 **Duration:** Week 3–4

📖 **Topics:**

- React Setup (Vite or Create-React-App)
- JSX, Components, Props, State
- Hooks: useState, useEffect, useRef
- React Router DOM

- Conditional rendering, lists/maps
- Tailwind utility-first design
- Type Script Basic in React
- Zod or Yup(for form validation)

❑ Mini Projects:

- Task Tracker App
- Multi-Page Blog UI
- Dark Mode Toggle Site

🌐 PHASE 3: MERN Stack Backend + Mongo

📅 Duration: Week 5–6

❑ Topics:

- Node.js + Express.js basics
- API Routes: GET, POST, PUT, DELETE
- MongoDB Atlas + Mongoose
- Middleware + Error Handling
- RESTful API patterns
- JWT Auth (optional, advanced)

❑ Projects:

- Blog API
- Notes App with DB
- CRUD Dashboard with Mongo

❑ PHASE 4: Python + AI/ML Basics

📅 Duration: Week 7–8

❑ Libraries:

- Python Syntax Refresher
- NumPy (arrays, indexing, reshape)
- Pandas (DataFrames, CSVs, cleaning)
- Scikit-learn (Regression, Classification, Model Evaluation)
- Saving Models using `joblib`

❑ Mini AI Projects:

- Text Input → Category Predictor
- CSV Upload → Classify Rows

- Model Training + Save Model
-

💻 PHASE 5: NLP + Transformers (Smart Text AI)

📅 Duration: Week 9–10

📖 Libraries:

- `nltk / spacy` (cleaning, tokenizing, lemmatizing)
- `transformers` (BERT, T5, etc.)
- `sentence-transformers` (semantic similarity)

💡 Projects:

- Resume Scoring App
 - Smart Form Autocomplete
 - Chatbot Response Sorter
-

🖼 PHASE 6: Image + Audio AI (Optional)

📅 Duration: Week 11–12

📖 Libraries:

- `opencv-python, PIL` → for image load/convert
- `librosa, soundfile` → for audio input to NumPy arrays

💡 Projects:

- Age Estimator from Image
 - Voice Mood Detector
 - Image Caption Generator
-

🌐 PHASE 7: Web + AI Integration (Frontend + Backend Merge)

📅 Duration: Week 13–14

📖 Concepts:

- FastAPI or Flask to serve AI model
- API → takes request → returns prediction (JSON)
- Connect React frontend to Python backend
- File Upload (text, image, audio)
-

⌚ Workflow:

React UI → API Request → AI Model → JSON Response → UI Output

🎁 Full Projects:

- AI Study Assistant
 - Smart Job Recommender
 - Voice-Powered Command App
-

🔧 PHASE 8: Deployment & Hosting

📅 Duration: Week 15–16

🔧 Frontend:

- Vercel / Netlify for React

🔧 Backend:

- Railway / Render / EC2
- .env, uvicorn, Procfile setup
- Test JSON response in production

🎁 Fullstack:

- Host React + Flask/FastAPI
- Connect domains (optional)
- Add loading states & error messages

2. DEVOPS + CLOUD + SYSTEM DESIGN

🎯 Goal:

To master modern DevOps practices, confidently work with cloud infrastructure, and build scalable system design knowledge that makes you ready for high-level technical interviews or SRE/DevOps roles.

🎁 Final Outcome:

- Hands-on DevOps toolchain knowledge
 - Real cloud deployments (AWS, Render, etc.)
 - Projects using CI/CD + Containerization
 - Scalable architecture understanding
 - Resume-worthy practical system design examples
-

📘 Full 3-Part DevOps + Cloud + System Design Roadmap (6 Weeks)

Suggested)

PHASE 1: Linux, Git, and Shell Scripting

Duration: Week 1

 Topics:

- Linux basics (CLI, file handling, users/permissions)
- Git workflows (init, clone, push, PRs)
- Bash scripting basics (loops, conditions, file ops)

Practice:

- Custom shell scripts
- Daily git push habit
- .sh automation mini tasks

PHASE 2: Docker + Containerization

Duration: Week 2

 Topics:

- What is Docker, why use it?
- Docker CLI (build, run, ps, images, exec)
- Dockerfile creation
- Docker Compose for multi-container apps

Projects:

- Containerize a Node.js or Python app
- Use Docker Compose to run backend + DB

PHASE 3: CI/CD + GitHub Actions

Duration: Week 3

 Topics:

- CI/CD Basics
- GitHub Actions (workflow.yml)
- Auto-deploy on push
- Testing & linting pipelines

Projects:

- Auto-deploy React app using GitHub Actions + Netlify
- Auto test + lint workflow setup

PHASE 4: Cloud Computing (AWS Focus)

Duration: Week 4

 Topics:

- Basics of Cloud & IaaS, PaaS, SaaS
- AWS Core Services:
 - EC2 (virtual machines)
 - S3 (storage)
 - RDS (managed DBs)
 - IAM (user roles)
 - Route 53 (DNS)

 Projects:

- Host Dockerized app on EC2
- Store uploads to S3
- Connect EC2 ↔ RDS with security groups

PHASE 5: System Design Basics

Duration: Week 5

 Topics:

- Latency, Throughput, Load Balancing
- Horizontal vs Vertical Scaling
- Caching (Redis/CDN), DB sharding
- Message Queues (RabbitMQ/Kafka)
- CAP Theorem, Consistency

 Diagrams:

- Design a URL Shortener
- Design Instagram Feed
- Design Scalable File Upload Service

PHASE 6: Combine All — Capstone Project

Duration: Week 6

 Tools & Tech:

- Full DevOps workflow
- Cloud + Docker + GitHub Actions
- Load-balanced, monitored system

 Final Project Ideas:

- Realtime Chat App (deployed on EC2 with Docker + RDS)
- CI/CD Setup for Full Stack App
- Cloud-resilient blog app with DB failover

Bonus Tools (Explore as Needed):

- Jenkins (CI/CD)
 - Kubernetes (if time permits)
 - Terraform (Infra as code, optional)
-

✓ Final Outcome:

You'll have **real hands-on cloud experience**, can confidently **talk DevOps in interviews**, and build apps that are **production-grade**, scalable, and maintainable.

3. DSA IN C++ (Core Interview Prep)

Basic Concepts:

- Input/Output, Loops, Conditions
- Functions, Recursion
- Arrays and Strings

Intermediate Topics:

- Linked List (Singly, Doubly, Cycle detection)
- Stack and Queue (Next Greater Element, Min Stack)
- Trees (DFS, BFS, Traversals, BST)
- Graphs (Adjacency List, DFS, BFS, Dijkstra)
- HashMaps, Sliding Window
- Two Pointers, Binary Search

Advanced Topics:

- Backtracking (Sudoku, N-Queens)
- Greedy Algorithms (Activity Selection, Huffman Coding)
- Dynamic Programming (Knapsack, LCS, LIS, MCM)
- Tries and Segment Trees (if time allows)

DSA Practice Strategy:

- Use LeetCode or GFG
- Follow a sheet (Striver's or Love Babbar's)
- Solve at least 300 questions
- Give weekly contests (LeetCode/Codeforces)

4. 6-MONTH WEEKLY PLAN OVERVIEW

Month	Focus Area	Key Goals
Month 1	Full Stack Frontend + DSA Basics	React + 50 DSA Problems
Month 2	Backend + Auth + Docker + Intermediate DSA	Build & Dockerize app
Month 3	Full DevOps + AWS + Continue DSA	CI/CD + Deploy on Cloud
Month 4	AI API + System Design Start + DSA	AI Project + LLD Concepts
Month 5	Polish Projects + Advanced System Design	Resume Ready + DSA 250+

Month 6 Apply + Resume + Mock Interviews

Target Jobs + GitHub Portfolio

Perfect! Here's a detailed list of **certifications, achievements, and skills** for each track to include in your resumes. These are selected to be:

- **Free or low-cost**
 - **Recognized by recruiters**
 - **Relevant to your syllabus and learning goals**
-

✓ 1. Full Stack Developer + AI Integration

(Core Technical Skills)

- HTML, CSS, JavaScript (ES6+)
- React.js, Tailwind CSS
- Node.js, Express.js
- MongoDB, REST API
- Python, NumPy, Pandas
- Scikit-learn, Transformers (HuggingFace)
- Flask or FastAPI
- Git & GitHub, Postman
- Deployment (Vercel, Netlify, Render)

Recommended Certifications

Name	Provider	Link
Meta Front-End Developer	Coursera (Free audit)	Link
IBM Full Stack Software Developer	Coursera	Link
Python for Everybody	Coursera (Dr. Chuck)	Link
AI for Everyone	Coursera (Andrew Ng)	Link
GitHub: Git & GitHub Crash Course	freeCodeCamp/YouTube	Watch

Projects/Achievements

- AI Study Assistant Web App
 - Smart Job Recommendation System
 - Full Stack Blog + Notes App with Auth
 - Resume Scorer using BERT
 - Participated in Hackathon XYZ (optional)
 - GitHub streak: 90+ days coding
-

⌚ 2. DevOps + Cloud + System Design

(Core Technical Skills)

- Linux CLI, Bash Scripting
- Git, GitHub, GitOps
- Docker, Docker Compose
- CI/CD (GitHub Actions)
- AWS EC2, S3, IAM, Lambda
- Terraform (basics)
- Kubernetes (Intro)
- Nginx, Load Balancing
- System Design Patterns (Caching, Scaling, Load Balancing)

📘 Recommended Certifications

Name	Provider	Link
Google Cloud Essentials	Qwiklabs	Link
AWS Cloud Practitioner	AWS SkillBuilder (Free)	Link
Docker Essentials	IBM on Coursera	Link
GitHub Actions for CI/CD	LinkedIn Learning	Link
System Design Primer (GitHub)	Open Source	Link

🏆 Projects/Achievements

- Deployed CI/CD pipeline for React + Flask App
- Dockerized full-stack AI app
- AWS EC2-hosted AI microservice
- Load Balanced Web App using Nginx
- Terraform Infra Setup (Mini)

□ 3. AI Integration + Python

💡 Core Technical Skills

- Python, NumPy, Pandas
- Scikit-learn, Matplotlib, Seaborn
- Hugging Face Transformers, BERT
- Sentence Transformers, NLTK
- Flask/FastAPI (serve models)
- CSV/JSON Data Processing
- REST API Integration with React

📘 Recommended Certifications

Name	Provider	Link
AI For Everyone	Coursera	Link
Machine Learning with Python	IBM on Coursera	Link
NLP with HuggingFace	HuggingFace Free Course	Link
FastAPI Crash Course	YouTube (freeCodeCamp)	Watch
Python for AI	Google	Link

🏆 Projects/Achievements

- Resume Scorer using BERT & Streamlit
- Text Classifier using joblib + Flask
- CSV Classifier + Dashboard
- AI chatbot using Transformers
- Contribution to open-source model (optional)

