Python Full Stack Developer with AI/ML Integration

Potential offerings of this course

- **High Demand in 2025 Job Market**: Full stack Python developers are highly sought after, with job postings rising 30% year-over-year, driven by Python's versatility in web development, AI, and cloud solutions.
- **Competitive Salaries**: Earn ₹5,00,000–₹20,00,000 annually in India, with global averages reaching $78,300–$125,000, especially for skilled developers in tech hubs.
- **AI/ML Integration**: Learn to build intelligent web apps using TensorFlow, Hugging Face, and pre-trained models, aligning with 2025's trend of AI-driven development.
- **Cloud-Native Expertise**: Master AWS, Azure, and Django to create scalable applications, meeting the growing need for cloud-based solutions.
- **Comprehensive Skill Set**: Gain proficiency in React, Django, PostgreSQL, and CI/CD pipelines, preparing you for versatile roles in startups and enterprises.
- **Hands-On Capstone Project**: Build a full stack app with AI features, creating a portfolio that showcases your skills to employers.
- **Beginner-Friendly Curriculum**: No prior experience needed; learn Python, front-end, and back-end development with practical tools like Jupyter Notebooks and Streamlit.
- **Future-Proof Career**: Stay ahead with skills in Progressive Web Apps (PWAs) and cybersecurity, critical for 2025's evolving tech landscape.
- **Flexible Learning**: Designed for students and professionals, with free tools like VS Code and Tailwind CSS, fitting busy schedules over 18 weeks.
- **Global Opportunities**: Leverage remote work trends, with Indian developers accessing international roles in AI, web, and cloud development.

Phase 1: ***Programming Fundamentals (4 weeks)***

Learning Outcomes

- ✓ Write clean, efficient Python code using proper syntax and best practices
- ✓ Implement object-oriented programming concepts in Python
- ✓ Create modular, reusable code with functions and classes
- ✓ Debug Python programs and manage dependencies
- ✓ Use Git and GitHub for version control

Tools Used

- ✓ Python 3.x (latest stable version)
- ✓ Visual Studio Code or PyCharm Community Edition
- ✓ Git and GitHub
- ✓ Command line interface (Terminal/PowerShell)
- ✓ Python Package Index (PyPI) and pip
- ✓ Jupyter Notebooks

❖ **Phase 2:** *Backend Development (6 weeks)*

Learning Outcomes

- ✓ Design and build RESTful APIs using Python frameworks
- ✓ Implement server-side logic and database operations
- ✓ Create and manage database schemas and relationships
- ✓ Develop authentication and authorization systems
- ✓ Deploy backend applications to production environments

Tools Used

- ✓ Flask and Django frameworks
- ✓ SQLite (development) and PostgreSQL (production)
- ✓ DBeaver (free database management tool)
- ✓ Postman or Insomnia (API testing)
- ✓ Django REST Framework
- ✓ SQLAlchemy ORM
- ✓ pytest for backend testing

❖ **Phase 3:** *Frontend Development (4 weeks)*

Learning Outcomes

- ✓ Create responsive web pages with HTML5 and CSS3
- ✓ Implement interactive user interfaces with JavaScript
- ✓ Build component-based applications using React
- ✓ Manage state and handle events in frontend applications
- ✓ Consume APIs and handle asynchronous operations

Tools Used

- ✓ HTML5, CSS3, JavaScript (ES6+)
- ✓ React.js
- ✓ Bootstrap or Tailwind CSS (free frameworks)
- ✓ npm or yarn package managers
- ✓ Chrome/Firefox DevTools
- ✓ React Developer Tools (browser extension)

❖ Phase 4: *Full Stack Integration (3 weeks)*

Learning Outcomes

- Connect frontend and backend components into cohesive applications
- Implement secure user authentication flows across the stack
- Configure basic CI/CD pipelines for deployment
- Optimize applications for performance
- Implement security best practices across the stack

Tools Used

- RESTful APIs
- JWT for authentication
- GitHub Actions (basic CI/CD)
- Nginx (web server)
- Heroku, Vercel, or PythonAnywhere (free tiers)

Phase 5: *AI/ML Fundamentals & Integration (2 weeks)*

Learning Outcomes

- Understand basic AI and ML concepts and terminology
- Implement simple data analysis and visualization
- Use pre-trained ML models in web applications
- Build simple AI-powered features for web applications
- Process and display model predictions in user interfaces

Tools Used

- NumPy and Pandas (basics)
- Matplotlib for simple visualizations
- scikit-learn for basic classification and regression
- Pre-trained models from Hugging Face
- Teachable Machine (Google's no-code ML tool)
- Streamlit for simple ML dashboards
- Flask/Django for model deployment

Phase 6: *Capstone Projects (2 weeks)*

Learning Outcomes

- Design, build, and deploy a complete full stack application with simple AI/ML features
- Work effectively in a team using agile methodologies
- Document code and create technical specifications
- Present and explain technical solutions to different audiences

Tools Used

- All previously introduced tools
- Trello or GitHub Projects (free project management)
- Figma (free tier for UI/UX design)
- Markdown for documentation