**Internship Program: Full Stack Development Roadmap**

This internship program is designed to provide a comprehensive understanding of software development, from foundational programming concepts to agile methodologies and practical project implementation. The curriculum is structured to equip interns with the skills necessary for effective team collaboration and real-world application development.

**Course Goals:**

* Understand the motivation behind programming and collaborative development.
* Gain proficiency in fundamental programming concepts and object-oriented principles.
* Learn about networking basics, APIs, and web servers.
* Master version control with Git and GitHub.
* Apply Agile Scrum methodologies in a team environment.
* Develop full-stack applications through project-based learning.
* Prepare for technical interviews and future career paths in software development.

**Week 1: Orientation and Fundamentals**

This week focuses on setting the stage for the internship, introducing core programming concepts, and initiating version control practices.

* **Day 1: Motivation and Introduction**
  + Why learn to code? The future of programming.
  + Brief introductions and program roadmap overview.
  + Goal setting for the internship.
* **Day 2: Basic Syntax of an Imperative Programming Language**
  + Introduction to programming fundamentals: Variables, Data Types, Basic Data Structures.
  + Control Structures: Conditions and Loops.
  + Simple coding exercises (e.g., building small applications like a phonebook, tic-tac-toe, or calculator).
* **Day 3: Functions and Methods**
  + Deep dive into functions and methods.
  + Concepts: Pass by Value vs. Pass by Reference, Deep Copy vs. Shallow Copy.
  + Introduction to Recursion.
  + Continue building existing small applications and discuss interview-relevant concepts.
* **Day 4: Data Persistence and Error Handling**
  + Storing and loading application data (e.g., in files).
  + Understanding and implementing Exception Handling.
* **Day 5: Introduction to Version Control (Git and GitHub)**
  + Understanding version control systems (Git and GitHub).
  + Creating repositories and backing up code.
  + Git branches, staging area, and commits.
  + Partial introduction to Agile Scrum and setting goals for the upcoming week.

**Week 2: Advanced Concepts and Agile Practices**

Building on the fundamentals, Week 2 introduces Object-Oriented Programming (OOP), networking essentials, and API concepts, culminating in a deeper dive into Agile Scrum.

* **Day 1: Introduction to Object-Oriented Programming (OOP)**
  + Core OOP Concepts: Encapsulation and Abstraction.
  + Practical implementation through examples (e.g., designing a class for a digital alarm clock).
* **Day 2: Advanced OOP Concepts**
  + OOP Concepts: Inheritance and Polymorphism.
  + Objects Association: Composition and Aggregation.
* **Day 3: Networking Basics**
  + Client, Server, Host, IP Address, Domain Name.
  + Interactive session on application communication across hosts.
  + Understanding HTTPS, URLs, and Web Servers.
* **Day 4: Introduction to APIs**
  + Consuming existing APIs (e.g., getting time of different regions).
  + Calling APIs for simple messages.
  + Using APIs to enhance applications.
  + Creating a basic API (calling a function on another host).
* **Day 5: Understanding Git Feature Branch Workflow and Scrum Deep Dive**
  + Contributing to existing code repositories via feature branches.
  + Methodologies for effective team collaboration.
  + Understanding pushing code and creating pull requests for code reviews.
  + Roles in Scrum (Product Owner, Scrum Master, Development Team).
  + Overview of Scrum Events and Artifacts.

**Weeks 3-8: Project-Based Learning (Sprints 1-3)**

The latter part of the internship is dedicated to hands-on project development within a team, following Agile Scrum sprints. Each team will work on a specific project, applying the concepts learned in the initial weeks.

* **Sprint Structure:**
  + Number of Sprints: 3
  + Duration of each Sprint: 10 days
* **Team Assignments (Examples):**
  + Batch 1: Alpha - Phone Book API
  + Batch 2: Beta - Anonymous Feedback API
  + Batch 3: Gamma - Motivational Quotes API
  + Batch 4: Delta - E-Sports Info API
* **Weekly Plan for Project-Based Learning:**
  + **Weeks 3-4 (Sprint 1): Backend Development & Data Persistence**
    - **Week 3:** Backlog refinement, sprint planning, project setup, initial backend Get/Create operations, dummy data for testing, Postman testing, daily Scrum.
    - **Week 4:** Persisting data in files, continued Postman testing, daily Scrum, preparation for Sprint Review, Sprint Review Meeting (API demo), Sprint Retrospective.
  + **Weeks 5-6 (Sprint 2): Feature Enhancement & UI Integration (Roadmap)**
    - **Week 5:** Backlog refinement, sprint planning, adding new API features (e.g., sorting, filtering), daily Scrum. Exploration of UI templates (e.g., Bootstrap) and modifying index.html.
    - **Week 6:** Continued feature development, daily Scrum, preparation for Sprint Review, Sprint Review Meeting (complete API demo), Sprint Retrospective. Roadmap includes adding CRUD forms and making them functional with JavaScript.
  + **Weeks 7-8 (Sprint 3): Testing, Deployment & Interview Preparation**
    - **Week 7:** Backlog refinement, sprint planning, bug identification, real-life mock interviews, daily Scrum. Testing all functionalities, bug fixing, creating and running application artifacts on other machines.
    - **Week 8:** Final application testing, daily Scrum, Sprint Review Meeting (demo of the entire application), Sprint Retrospective. Roadmap includes installing a web server and deploying the application. The final day includes a discussion of the overall roadmap, sharing interview preparation guidelines, and feedback on team progress and interviews.

**End of Internship Training**

The program concludes with a comprehensive review of covered topics, a detailed roadmap for continued learning, and final mock interviews to prepare interns for their professional journey.