**Full Stack Development - 5-Day Internship Plan**

**Project:** Portfolio Website

Each day includes:

* **Concepts & Explanation**
* **Hands-on Coding Session**
* **Practical Exercises**

**🔹 Day 1: Introduction to Web Development & Frontend Basics**

🎯 **Goal:** Build the basic structure of the portfolio website

**Session 1: Introduction to Full Stack Development**

* Overview of **Frontend, Backend, and Database**
* How websites work (Client-Server Model, HTTP Requests)
* Introduction to **HTML, CSS, JavaScript**

**Session 2: Setting Up the Project**

1 Install **VS Code, Node.js, GitHub**  
2 Create a project folder:

mkdir portfolio-website && cd portfolio-website

3 Create initial files:

* index.html, style.css, script.js

**Session 3: Build the HTML Structure**

* Create **Header, About, Projects, Blog, and Contact sections** in index.html
* Hands-on:
  + Add profile information
  + Create a simple **navigation bar**

✅ **Homework:** Improve page styling using CSS

**🔹 Day 2: Advanced Frontend - JavaScript & GitHub**

🎯 **Goal:** Add interactivity + push project to GitHub

**Session 1: JavaScript & DOM Manipulation**

* Variables, Functions, Event Listeners
* Handling **form inputs & button clicks**

**Session 2: Git & GitHub for Version Control**

1 **Initialize Git**

git init

git add .

git commit -m "Initial Commit"

git branch -M main

2 **Push to GitHub**

git remote add origin <your-github-repo-url>

git push -u origin main

**Session 3: Adding JavaScript to Portfolio**

* Implement form validation in script.js
* Hands-on: Add a **"Projects"** section using JavaScript
* Add **"Blog"** section layout in index.html
* <section id="blog">
* <h2>Blog</h2>
* <div id="blog-list"></div>
* </section>

✅ **Homework:** Research Bootstrap or TailwindCSS for styling

**🔹 Day 3: Backend Basics - Node.js & Express**

🎯 **Goal:** Set up a backend for handling project data, blog posts & contact form

**Session 1: Introduction to Backend & API Development**

* What is **Node.js & Express.js**?
* Setting up a server

**Session 2: Installing & Setting Up Express**

1 Install dependencies:

npm init -y

npm install express cors nodemon

2 Create server.js and write basic API routes

**Session 3: Fetching Data from Backend**

* Implement /projects API endpoint
* Implement /blog API endpoint
* let blogs = [
* { id: 1, title: "My First Blog", content: "This is my first blog post." },
* { id: 2, title: "Web Development Basics", content: "Learn HTML, CSS, and JavaScript." }
* ];
* app.get("/blog", (req, res) => {
* res.json(blogs);
* });
* Hands-on: Modify script.js to **fetch blog data dynamically**
* fetch("http://localhost:5000/blog")
* .then(response => response.json())
* .then(data => {
* let blogList = document.getElementById("blog-list");
* data.forEach(blog => {
* let div = document.createElement("div");
* div.innerHTML = `<h3>${blog.title}</h3><p>${blog.content}</p>`;
* blogList.appendChild(div);
* });
* });

✅ **Homework:** Modify server.js to accept new blog posts

**🔹 Day 4: Database Integration & Deployment**

🎯 **Goal:** Store contact form & blog posts + deploy frontend

**Session 1: Introduction to Databases**

* Basics of **MongoDB / Firebase**
* Storing user inputs

**Session 2: Storing Contact Form & Blog Posts**

1 Modify server.js to handle POST request for contact form  
2 Modify server.js to accept **new blog posts**

app.post("/blog", (req, res) => {

const newBlog = req.body;

blogs.push(newBlog);

res.json({ success: true, message: "Blog post added!" });

});

3 Modify frontend to **send new blog post data to the server**

document.getElementById("add-blog").addEventListener("submit", function(event) {

event.preventDefault();

let title = document.getElementById("blog-title").value;

let content = document.getElementById("blog-content").value;

fetch("http://localhost:5000/blog", {

method: "POST",

headers: { "Content-Type": "application/json" },

body: JSON.stringify({ title, content })

})

.then(response => response.json())

.then(data => alert(data.message))

.catch(error => console.log("Error:", error));

});

* **Hands-on:** Test form submissions

**Session 3: Deploying the Frontend**

1 **Host on GitHub Pages / Netlify**  
2 Hands-on: Deploy & test

✅ **Homework:** Research deployment for backend services

**🔹 Day 5: Final Project, Debugging & Deployment**

🎯 **Goal:** Deploy backend & complete the portfolio website with blog

**Session 1: Deploying Backend on Render**

1 Push backend to GitHub  
2 Deploy on **Render.com**  
3 Test API endpoints

**Session 2: Adding Final Touches**

* Implement **Contact Form Email Integration**
* Improve **UI with animations**

**Session 3: Final Project Showcase & Certification**

* Debugging & Optimizations
* **Students present their portfolio websites**
* Feedback & Certification

**🎯 Final Outcome**

* Fully functional **Portfolio Website** with:
* **Frontend (HTML, CSS, JavaScript)**
* **Backend (Node.js, Express.js)**
* **Blog Section (View & Add Blog Posts)**
* **Database (MongoDB / Firebase)**
* **Deployed on GitHub & Render**