Day	Topic	Theory	References/Links	Practice-1	Practice-2	Trainer	Training Date
Day 1	Linux/Ubuntu & Git	Linux: - Overview of Linux distributions - Terminal navigation (cd, ls, pwd, mkdir, rmdir) - File handling (cp, mv, rm, touch, cat) - User and permissions (chmod, chown, sudo) - Package management (apt-get, snap) Git: - What is version control? - Git architecture (working directory, staging area, repo) - Git commands: init, clone, status, add, commit, log - Remote repositories, GitHub, SSH setup - Branching, merging, resolving conflicts		- Practice terminal commands: navigation, file operations - Set up user permissions and manage processes - Install packages using apt-get - Initialize Git repo and commit changes - Create and merge branches locally - Push to and pull from GitHub - Resolve a merge conflict - Daily assignment: Document commands used in a README	- Write a bash script to automate file backup - Create and change file permissions for multiple files - Monitor running processes and write output to a file - Schedule a cron job to run a script every hour	Mahesh/Bharani	28-May
Day 2	SDLC - Agile & Kanban	SDLC: - Stages of SDLC (Requirement, Design, Development, Testing, Deployment, Maintenance) - Waterfall vs Agile Agile: - Agile Manifesto principles - Scrum roles (Scrum Master, Product Owner, Team) - Scrum ceremonies (Daily stand-up, Sprint planning, Retrospective) Kanban: - Workflow visualization - WIP limits, swimlanes - Tool usage (Jira, Trello)		- Create a sample project plan using SDLC stages - Set up Trello or Jira Kanban board - Add backlog items and assign team roles - Simulate a daily stand-up meeting - Conduct a mock sprint planning session - Daily assignment: Document differences between Agile and Waterfall	- Create a new Git repo and manage branches - Simulate a team collaboration workflow (clone, push, pull, merge) - Resolve a merge conflict manually - Use `.gitignore` to exclude specific files from tracking	Prakash	29-May
Day 3	Web & HTML	Web Basics: - Client-server architecture - HTTP/HTTPS protocols - DNS, IP, and domain names - Browser rendering process HTML: - HTML5 structure and semantics - Headings, paragraphs, links, lists - Images, audio, video tags - Forms and input elements - Semantic tags (header, footer, nav, article, section)		- Build a basic HTML resume/portfolio page - Create navigation with anchor links - Insert images and videos into the page - Build an HTML form (text, radio, checkbox, submit) - Use semantic tags to structure a blog layout - Daily assignment: Submit a structured HTML page	- Create a user story and acceptance criteria document - Simulate a 2-day sprint with 4 tasks using a F5- Create a personal resume webpage with sections for bio, projects, contact - Create a table with employee data and use proper table tags - Build a registration form with input validation (HTML5 types) - Design a basic blog post layout using semantic HTML board - Conduct a mock sprint review and retrospective - Create a burndown chart from a sprint	Vignesh Rathinam	30-May
Day 4	CSS	CSS Basics: - Syntax, selectors, specificity - Types of CSS (inline, internal, external) - Colors, fonts, background, text properties - Box model (margin, border, padding, content) - Display properties (block, inline, inline-block) - Positioning (static, relative, absolute, fixed) - Flexbox and Grid basics - Responsive design and media queries		- Style HTML resume with colors, fonts, and spacing - Apply box model adjustments (padding, margin) - Build responsive layout using Flexbox - Use media queries to adjust for mobile - Create a simple CSS animation (hover effect) - Daily assignment: Upload styled responsive form	- Style a login form using Flexbox - Create a responsive navigation bar - Implement dark/light theme switching using classes - Create a grid-based image gallery with hover effects	Mani.T	2-Jun
Day 5	SQL	Database Basics: - What is a database? - Relational vs non-relational databases - Tables, rows, columns SQL: - Data types and constraints - CRUD operations: CREATE, INSERT, SELECT, UPDATE, DELETE - Filtering and sorting: WHERE, ORDER BY, LIMIT - Aggregations: COUNT, SUM, AVG, GROUP BY, HAVING - Joins: INNER, LEFT, RIGHT, FULL - Subqueries and nested queries		- Install mysql - Create tables for users, products, orders - Insert sample data into tables - Run SELECT queries with filters and sorting - Perform JOIN operations between tables - Write aggregation queries using GROUP BY - Daily assignment: Share SQL file with queries run	- Design a database for a bookstore with tables: books, authors, customers, orders - Write queries to fetch top 5 selling books - Use JOIN to list customer orders with book details - Use GROUP BY to count number of books per author	Vijeyandran	3-Jun
Day 6	JavaScript	JavaScript Basics: - Introduction to scripting - Data types, variables (let, const, var) - Operators, conditionals (if, else, switch) - Loops (for, while, do-while) - Functions (declaration, expression, arrow functions) - Events and DOM manipulation - Arrays and objects - Basic debugging and console.log		- Write a script to validate HTML form fields - Build a counter app using DOM manipulation - Create a dynamic to-do list (add/delete items) - Add event listeners to buttons and inputs - Manipulate DOM elements with JS (change styles/text) - Daily assignment: Deploy a JS-powered page	Create a simple calculator (add, subtract, multiply, divide) - Build a dynamic to-do list with local storage - Validate a contact form using JS (email, phone number) - Build a color picker tool that changes background color	Gopinath	4-Jun
Day 7	Deployment - Docker & Server	Deployment Concepts: - What is deployment? Local vs production - Server basics: IP, domain, ports - Web servers (Apache, Nginx overview) Docker: - What is containerization? - Docker architecture (images, containers, volumes) - Writing a Dockerfile - Running and managing containers - Docker Compose basics - Introduction to CI/CD		- Install Docker and run Hello World container - Write a Dockerfile for the portfolio app - Build Docker image and run container - Expose port and access app via browser - Use Docker Compose for multi-container setup - Optional: Host containerized app on free cloud platform (like Render or Railway) - Daily assignment: Share GitHub repo with Docker setup	- Write a Dockerfile for a Node.js or Python Flask app - Containerize a simple HTML-CSS-JS app and expose on port 8080 - Use Docker Compose to run a web server and database together - Push Docker image to Docker Hub and pull it on another machine	Bharani	5-Jun