1. Building and Deploying a Static Website with JAMstack

- Definition: Introduce JAMstack (JavaScript, APIs, and Markup) as a modern architecture for building fast and secure static websites.

- Practical Implementation: Walk through building a personal portfolio site using JAMstack, focusing on static site generation with frameworks like Gatsby or Next.js.

- Approximated Time: 3-5 hours

1. Automating Data Pipelines with Python and Apache Airflow

- Definition: Discuss how data pipelines automate the movement and transformation of data for analytics and processing.

- Practical Implementation: Show how to use Apache Airflow with Python to schedule and automate data tasks like ETL (Extract, Transform, Load).

- Approximated Time: 5-7 hours

1. Automating Web Scraping with Python and BeautifulSoup

- Definition: Introduce web scraping as a technique for extracting data from websites for analysis or automation purposes.

- Practical Implementation: Show how to use Python with BeautifulSoup to scrape data from a website and store it in a CSV or database.

- Approximated Time: 3-5 hours

1. Handling Payments in Web Applications with Razorpay API

- Definition: Explain how Razorpay API enables developers to easily integrate payment processing functionality in web applications, allowing users to make secure online payments.

- Practical Implementation: Walk through the process of integrating Razorpay's payment gateway into a web application for handling transactions. The seminar will cover setting up the Razorpay API, creating payment orders, and handling payment responses, including success and failure scenarios, in a secure manner.

- Approximated Time: 4-5 hours

1. Web Scraping with Selenium for Dynamic Content

- Definition: Web scraping allows you to collect data from websites, and Selenium is a powerful tool for scraping dynamically generated content from modern web pages.

- Practical Implementation: Walk through scraping data from dynamic websites using Selenium WebDriver in Python or JavaScript.

- Approximated Time: 4-6 hours

1. Building a Task Scheduler with Node.js and Cron Jobs

- Definition: Explore how to automate task scheduling using Cron jobs and how to implement them in a Node.js environment.

- Practical Implementation: Build a task scheduler that runs specific tasks at regular intervals using Node.js and the `node-cron` package.

- Approximated Time: 3-4 hour

1. Building a Voice Assistant with Python

- Definition: Explore how to build a simple voice assistant capable of performing basic tasks such as setting reminders, searching the web, or controlling system functions.

- Practical Implementation: Build a Python-based voice assistant using libraries like SpeechRecognition and pyttsx3 for text-to-speech functionality.

- Approximated Time: 5-7 hours

1. Developing a Virtual Whiteboard for Team Collaboration

- Definition: Build an online whiteboard that allows team members to collaborate in real time on drawings, annotations, and brainstorming.

- Practical Implementation: Develop an interactive whiteboard using JavaScript libraries (e.g., Fabric.js) and enable real-time collaboration using WebSockets.

- Approximated Time: 6-7 hours

1. Creating an Online Voting System with Blockchain for Transparency

- Definition: Implement a secure and transparent voting system using blockchain technology to ensure vote integrity.

- Practical Implementation: Develop a decentralized voting system where votes are stored on the blockchain, ensuring transparency and security.

- Approximated Time: 6-8 hours