**VAP Curriculum**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Course code: | | Course title: **REST API Design and Fundamentals** | | |
| School / Centre: | |  |  |  |
| Beneficiaries: UG / PG/ Research | | UG |  |  |
| Course objective: | | REST API’s helps build a client-friendly distributed systems that are simple to understand and simple to scale. With this course learn to build fast and scalable API’s the right way. Understand the nuances of building an API: Right from the structure of an API to the best way to implement. Students will use node.js to implement and consume the API’s by the end of the course. | | |
| Course outcome: | | 1. Gain basic understanding of API Fundamentals 2. Appreciate API design guidelines 3. Implement API Patterns for better API design and consumption 4. Implement Hypermedia clients using API’s | | |
| S. No. | Topic | | | Duration (hours) |
| 1 | JavaScript – Recap   1. JavaScript Data Types – Primitive and Object 2. Arrays 3. Functions   Hands on node.js   1. Getting started with node.js 2. First Node Program 3. Concepts of Event Loop and How node.js works 4. Basics of Node Module Systems 5. Callbacks 6. Use built-in modules: File, Path, Events 7. Implement a Web Server with HTTP module | | | 6 |
| 2 | Node package manager   1. Installation and Usage 2. Package Dependencies 3. Semantic Versioning   Hands on with utility packages   1. lodash.js 2. nodemon | | | 2 |
| 3 | Express.js (Web Application Framework)   1. First WebServer Application 2. Handling GET, POST HTTP requests 3. Creation of Routes 4. Middleware 5. Templates 6. HTTP Headers | | | 3 |
| 4 | Introduction to Promises and Async programming | | | 3 |
| 5 | Mid-Term Mini Project: Build a mock node Web Application for a site like Udemy that comprises of course, students and lectures. | | | 3 |
| 6 | API’s –   1. General Introduction 2. Concepts and Terminology of API’s 3. Economics of API 4. Introduction to Origin of REST API 5. Introduction and Setting up POSTMAN | | | 2 |
| 7 | REST API Fundamentals   1. Resources 2. Universally accepted Design constraints for REST API’s | | | 2 |
| 8 | REST Resources   1. Singular 2. Plural 3. CRUD Mapping for REST | | | 2 |
| 9 | REST Security   1. OAUTH 2. JSON Web Token 3. Rate Limiters | | | 2 |
| 10 | REST API Tools   1. Introduction to Swagger 2. Swagger based API Design 3. Client SDK generators 4. Documentation generators | | | 3 |
|  | Final Project  Build REST API’s for the Mid Term Mini Project. The REST API’s should have RESOURCE identification for the various functionalities and consumption of the API’s for the site. Evaluation criteria will be based on API design, identified RESOURCES, API Documentation | | | 3 |
|  | Total course duration | | | 31 |