**Course Outline Template**

|  |  |
| --- | --- |
| **Please furnish the details in the below format and submit the form to the TD SPOC for review**  **The expectation is that the course should include 30% theory and 70% hands-on exercises** | |
| **Course Name** | Building Web applications using Node JS |
| **Course Duration** | 5 days (20 hours) |
| **Course Timings** | 2:00:00 PM to 6:00:00 PM |
| **Course Start Date** | 1st Feb 2021 |
| **Course End Date** | 5th Feb 2021 |
| **Mode of Training** | Webinar |
| **Webinar Link** | **IIHT will share** |
| **Hardware Requirement required for the participant** | RAM- 8GB, Windows OS  Hard Disk-120GB |
| **Software Requirement required for the participant** | **LIST OF SOFTWARE TO BE INSTALLED BEFORE TRAINING STARTS**   1. Git CLI on participant systems and GitHub account for every participant (to be created individually by participant).   Git CLI download: <https://git-scm.com/downloads>  GitHub link for account creation: <https://github.com/join?source=header-home>  Once an account is created by everyone, **the list of GitHub user names needs to be shared with me** - I will add them to the GitHub repository before start of training.   1. Node.js needs to be installed on all systems – Mac OSX, Linux and Windows is supported. The 10.x.x (LTS version) may be installed. This will also install npm. However, the proxy server details may need to be configured to enable npm access the npm registry (this registry is required to download Node modules required to build Node applications) – **the proxy configuration is a necessary step and has to be completed before training starts**.   Node.js <https://nodejs.org/en/download/>  To configure the proxy for npm these articles will be helpful:  <https://jjasonclark.com/how-to-setup-node-behind-web-proxy/>  <https://forum.freecodecamp.org/t/npm-behind-a-proxy-server/19386>   1. Depending on choice of database for the training, one of MongoDB or MySQL needs to be installed. 2. **MongoDB** needs to be installed on all systems. Instructions to download and install on various platforms can be found in the links below.   <https://www.mongodb.com/download-center?jmp=nav#community>  <https://docs.mongodb.com/manual/administration/install-community/>  Additionally, a \data\db (on root drive where MongoDB is installed on Windows, say C:\ or D:\) or /data/db folder (on Linux / Mac OSX) is required to be created. Participants should have write permissions on this folder.  Next download and install **Robo 3T** - <https://robomongo.org/download>   1. **MySQL** – Download and install from <https://dev.mysql.com/downloads/mysql/>   Next download and install **MySQL Workbench** from <https://dev.mysql.com/downloads/workbench/>   1. **PostgreSQL** – Download and install from <https://www.postgresql.org/download/>   Next download and install **pgAdmin 4** from <https://www.pgadmin.org/download/>   1. Download and install **VSCode** - <https://code.visualstudio.com/download> 2. For browser – latest version of **Chrome** is preferred. Internet Explorer is not acceptable.   Chrome: <https://www.google.com/chrome/browser/desktop/index.html>  **Additionally, it would be great if participants have as little restrictions (as permissible) on internet access during the session** |
| **Faculty Name** | Arun Vijayarengan |
| **Faculty Profile** | Attached |
| **Course Pre-requisites** | * Working knowledge of HTML * Very good knowledge of JavaScript – functions, objects, closures, passing and returning functions, callback functions. * However knowledge of ES2015+ (ES6+) features is NOT required.   Basic knowledge of web application concepts is a plus, but not necessary |
| **Course Objectives** | * Learn why server-side JavaScript is useful * Install Node.js * Learn how Node.js is architected to allow high scalability with asynchronous code * Create basic web applications with Node.js * Automate tasks with Gulp * Build an HTTP server using the core modules in Node.js * Use stream I/O to efficiently serve the web pages * Create modules to organize the server * Test the reliability of the application with unit tests * Convert the application to an MVC framework using Express * Interface to a MongoDB database and a web service. |
| **Agenda - Day 1** | * **Introduction**   + What is NodeJS?   + Why NodeJS?   + Architecture of NodeJS * **Setting up NodeJS**   + Installing NodeJS   + Understanding NodeJS console   + Understanding how JS differs on the front-end and back-end   + NPM and Various NPM commands   + First Node App * **V8 JavaScript Engine and Node Core** * **Essential ES6/ ECMAScript 2015 for Node JS Developers**   + Difference between JS5 / ES2015   + let and const   + Template Literals   + Arrow / Fat arrow Functions   + Rest and Spread operators   + Introducing Classes in JavaScript   + Modules in ES6 (import and export)   + Promises   + De-structuring in Objects and Arrays * **Modules, Exports, and Require**   + Creating our own Module   Module - *export* and *require* |
| **Agenda – Day 2** | * **Events and Streams**   + Events   + Event Emitter   + Event Loop   + Non-Blocking Asynchronous Execution   + Streams and Buffers in Node JS   + Pipes * **Accessing File System**   + Process Object   + Interacting with the file system   + OS Module * **Interacting with the Web**   + Making web requests in NodeJS   Web server in NodeJS |
| **Agenda – Day 3** | * **Using ExpressJS Framework**   + Starting Express App   + Understanding Middleware   + RESTful API’s and JSON   + Router   + Handling HTTP Requests   + Dealing with GET, POST, PUT, DELETE   + Request Object   URL Params, Query Params, Req Body and Res object |
| **Agenda – Day 4** | * **MongoDB**   + Understanding MongoDB   + Understanding Mongoose   + Understanding Schema   Queries in MongoDB and Mongoose |
| **Agenda – Day 5** | * + Storing Data   + Accessing Data   + CRUD in MongoDB   + Integrating MongoDB with ExpressJS App   + Completing REST API * Debugging - Industry Standard Techniques * **Connecting to MySQL / PostgreSQL** * Fixing CORS Errors * Quick Intro to Authentication |