Development Environment Document

for

TimeTable Generator

CS-08

September 26, 2018

Indian Institute of Information Technology Vadodara

Team Members

Aman Yadav (201651007)

DakshKumar Gondaliya (201651014)

Kirtika Singhal (201651024)

Mayank Pathela (201651029)

Nikhil Sachan (201651034)

Parmeshwar Kumawat (201651035)

Revision History

Version	Date	Name	Description
1	02/10/2018	Mayank Pathela	Initial Document
2	09/10/2018	Daksh Gondaliya	Review

Table of Contents:

Setting-up the environment for development:-

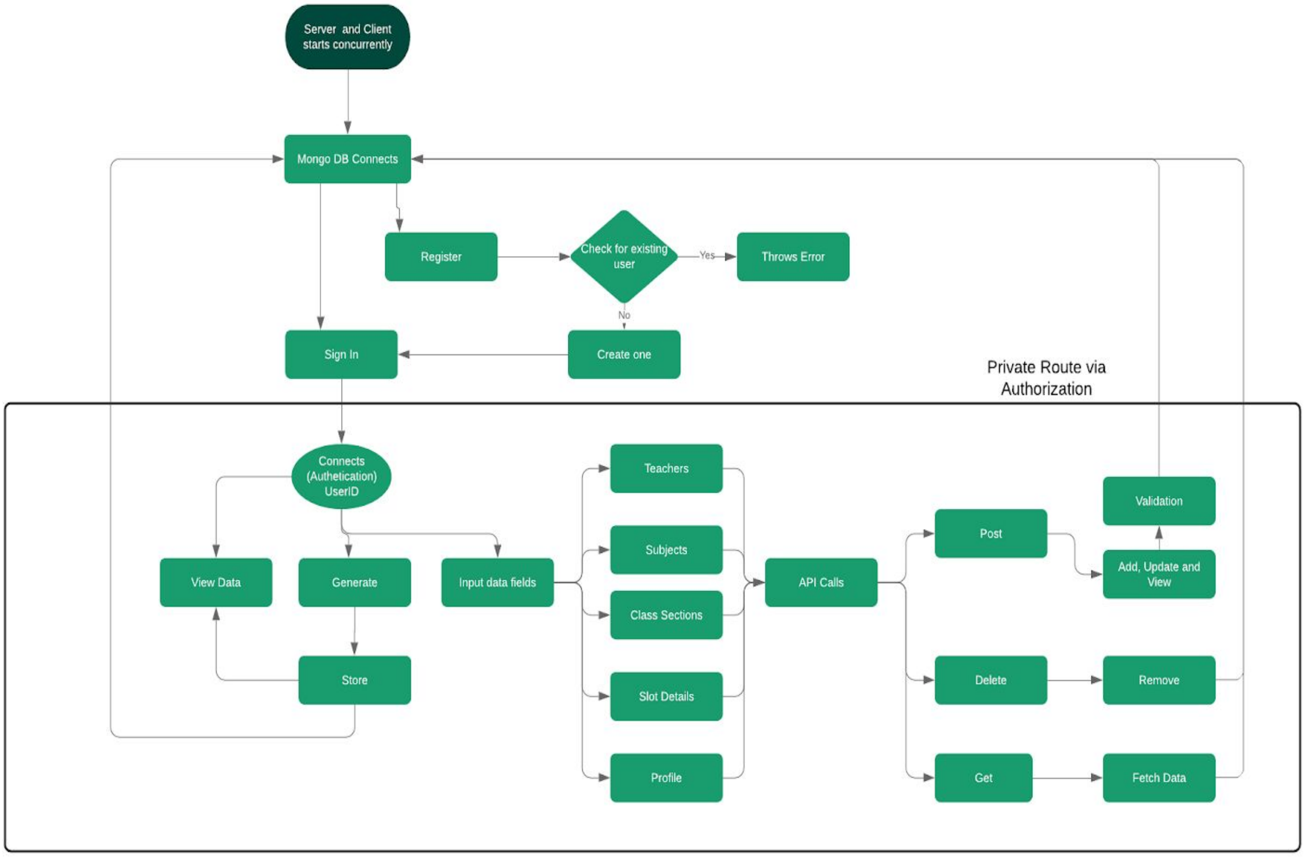
1. Install nodejs - https://nodejs.org/en/download/
2. Install Visual Studio Code - https://code.visualstudio.com/download/
3. Create a Project folder then navigate inside the folder and create client folder which will contain all of our front-end files.
4. Install create react-app
    - Type this command - `npm install -g create-react-app` in terminal.
    - To initialize the react-app type `create-react-app app-name` in terminal.
5. Install postman - https://www.getpostman.com/apps/
6. Create a new database on mlab.
7. Install Google Chrome browser with react and redux chrome extension.
8. O.S: Windows 10 or Higher, RAM:- 4GB or higher.

Usage of different technologies and platform:-

- Nodejs will be used for creating our back-end side server which will use MongoDB to store our data. We will be using mlab AWS-Sandbox cloud storage to store all of our user data.
- Visual Studio Code is a lightweight but powerful source code editor with integrated terminal which runs on our desktop and is available for Windows, macOS and Linux. It comes with built-in support for JavaScript, TypeScript and Node.js and has a rich ecosystem of extensions for other languages (such as C++, C#, Java, Python, PHP, Go) and runtimes (such as .NET and Unity).
- React is a JavaScript library for building interactive user interfaces. We can build encapsulated components that manage their own state, then compose them to make complex UIs.
- Postman allows us to create collections of integration tests to ensure your API is working as expected. Tests are run in a specific order with each test being executed after the last is finished. For each test, an HTTP request is made and assertions written in javascript are then used to verify the integrity of our code. Since the tests and test assertions are written in JavaScript, we have freedom to manipulate the received data in different ways, such as creating local variables or even creating loops to repeatedly run a test.
- JWT-authentication to manage different user and their data. Once the user is logged in, each subsequent request will include the JWT(JSON Web Token), allowing the user to access routes, services, and resources that are permitted with that token.
- Express is a minimal and flexible Node.js web application framework that provides a robust set of features for web and mobile application. We use routing using methods of the Express app Object that corresponds to HTTP methods.

- We will use nodemon tool that helps develop node.js based applications by automatically restarting the node application when file changes in the directory are detected.And to concurrently start our back-end and front-end we will use `"concurrently \"npm run server\" \"npm run client\"".`
- Redux and React chrome extension are used for analysing states.

Workflow Diagram:-


Server and Client starts concurrently
Mongo DB Connects
Register
Check for existing user
Yes
Throws Error
No
Create one
Sign In
Private Route via Authorization
Connects (Authetication) UserID
View Data
Generate
Store
Input data fields
Teachers
Subjects
Class Sections
Slot Details
Profile
API Calls
Post
Validation
Add, Update and View
Delete
Remove
Get
Fetch Data