

Hospital API with Unit Testing

Test Name: Hospital API tests

Time: 8 - 10 hours

COMPLETE/FIX YOUR HOSPITAL API CODE FROM THE PAST FEEDBACK. IF YOU ARE DOING THIS FOR THE FIRST TIME, THEN WRITE IT WELL!

- Design the server side for a hospital **Only the API needs to be designed**
- Tech Stack: Node.js & Mongo DB
- Feel free to use any libraries (you should be very clear on how it works conceptually)
- **Write Unit Tests for the below mentioned API's**

Extra Points:

- Well commented code
- Readme on how to setup the project on local system
- Scaleable folder structure (separate models, controllers and routes)

Task (same as Hospital API)

- Make regular commits to your code. Your submission will be rejected if you don't do this.

- Theme:
 - We're going to design an API for the doctors of a Hospital which has been allocated by the govt for testing and quarantine + well being of COVID-19 patients
 - There can be 2 types of **Users**
 - Doctors
 - Patients
 - Doctors can log in
 - Each time a patient visits, the doctor will follow 2 steps
 - **Register** the patient in the app (using phone number, if the patient already exists, just return the patient info in the API)
 - After the checkup, create a **Report**
 - Patient Report will have the following fields
 - Created by doctor
 - Status (You can use enums if you want to):
 - Can be either of: [Negative, Travelled-Quarantine, Symptoms-Quarantine, Positive-Admit]
 - Date
- Required Routes
 - /doctors/register → with username and password

- /doctors/login → returns the JWT to be used
- /patients/register [Write Tests to check that the API returns the newly created patient]
- /patients/:id/create_report [Write Tests to check that the API returns the newly created report]
- /patients/:id/all_reports → List all the reports of a patient oldest to latest [Write Tests to check that the API returns the newly created report]
- /reports/:status → List all the reports of all the patients filtered by a specific status
- Decide the fields and schemas on your own, smartly
- Decide which routes need to be protected by authentication

Free Hints

- **YOU CAN USE 'MOCHA' AND 'CHAI' FOR WRITING TESTS**
 - Some resources to begin:
 - <https://www.sitepoint.com/unit-test-javascript-mocha-chai/>
 - <https://scotch.io/tutorials/test-a-node-restful-api-with-mocha-and-chai>
 - www.google.com
- **USE PEN AND PAPER TO CREATE SCHEMAS BEFORE WRITING CODE**
- **WORK ON ONE API AT A TIME, DON'T THINK ABOUT THE COMPLETE APP TOGETHER**

To Submit:

- Push your code to a github repository
- Record a video (max 6 minutes) with the following details:
 - Explaining directory structures
 - Taking through each working route on Postman
 - Explain the code written for Tests
 - Running each Test and explaining how it works
- Upload the video to Youtube or Drive
- Submit with this form: <https://forms.gle/wfjmufaipc257Coj6>