# Full Stack Mock Test

LEVEL 5
Deadline: Monday, 7th Sept midnight (Ideal time of completion - 15hrs)

A company constantly needs to download their data to compile different reports. You need to create an interface for the employees of this company to fill in the data into the database and then download it in CSV format.

Create a complete web application, not an API

If you don't know what's a CSV, google it.

**Extra Points:**
- Well commented code
- Readme on how to setup the project on local system
- Scaleable folder structure (separate models, controllers and routes)
- Clean looking pages

**Task**
- Team Career Camp wants to maintain a database of all the student interviews [only for their own use], for this they store the following details:
    - Batch
    - Student Details (including college, status: [placed, not_placed])
    - Course Scores (including DSA Final Score, WebD Final Score, React Final Score)
    - Interviews (including company name and Date)
    - Results (this is a mapping between company, and student, contains result: [PASS, FAIL, On Hold, Didn't Attempt])
- Pages
    - Sign Up and Sign In only for employees
    - List of students + add new student (this is similar to adding and viewing posts in codeial)
    - List of Interviews + form to create an interview (with date)
        - Allocate a student to an interview
        - Select an interview to view the list of all students and mark a result status from the list page itself
    - [BONUS FEATURE] External Jobs List:

- Create a page which fetches real available jobs in India for react/node.js. Find and use open APIs [example: https://jobs.github.com/api, find at least one more, you can cheat!]
            - The API request should be handled from your node.js server
            - The design of this page should be minimalistic
            - Link to apply would be an external link, so don't worry about creating models/controllers/views for any part on this page
    - Download a complete CSV of all the data with the following columns:
        - Student id, student name, student college, student status, DSA Final Score, WebD Final Score, React Final Score, interview date, interview company, interview student result
        - A student can have multiple entries based on the interviews she/he has given.

**Free Hints**
    - YOU CAN FIND A GOOD NPM LIBRARY TO READ/PARSE/CREATE CSV
    - YOU CAN USE BOOTSTRAP/OTHER CSS FRAMEWORK TO MAKE YOUR APPLICATION LOOK BETTER, BUT IT'S NOT REQUIRED TO BE BEAUTIFUL
    - **USE PEN AND PAPER TO DEFINE THE STEPS**
    - WORK ON ONE STEP AT A TIME (in the order they are mentioned, DON'T THINK ABOUT THE COMPLETE APP TOGETHER

Create a complete web application, either by creating APIS in node and frontend in React or a server side node app with ejs templates

**To Submit:**
    - Record a video (max 3 minutes) with the following details (**and upload it to drive/youtube with public access, VIDEO SHOULD BE ACCESSIBLE BY ANYONE**):
        - Explain through the folder structure (what is placed where and why)
    - Upload the project files on github.com
    - Fill the github and video links in this form: https://docs.google.com/forms/d/e/1FAIpQLSdirrvDO6Ya-5nOkjGClhYe57rmv-yntVK0e5D4SRm_gkhp5Q/viewform?usp=sf_link