Use the following for development: Python, fastapi, pydantic, SQLAlchemy, postgresql

Use the following for testing: POSTMAN APIs

Output: POSTMAN collection to test the APIs and FastAPI server running

You need to complete all the 5 tasks mentioned below

Notes to aspirants:

* You can practise the above tasks at home and come to our office to take live test.
  + You cannot use code developed at home, you have to write again from beginning.
  + You can use chatgpt and internet also to assist you but should not copy the code from other sources. Each line you must type and write.
  + copying code already written at home, will disqualify you.
* You must complete all the above tasks in 6 hr to 8 hr time
* There may be small variations to the above questions during actual test
* You will be evaluated on the following parameters
  + coding standards followed - code structure, code quality, exception handling, test cases, logging, config files handling etc.,
  + ability to explain your work and code walk through
  + communication skills (telugu/hindi speaking is also ok)
  + ability to understand and solve problems
* confirm to us once you are ready to take the live test at our office, we will arrange the same. (Lunch is provided by us during programming test)

**Programming Test**

Create a database with the following schema

table institute (

column institute\_id - int - auto increment primary key NOT NULL

column institute\_name - varchar(100) - unique NOT NULL

)

table course (

column course\_id - int - auto increment primary key NOT NULL

column institute\_id - int - foriegn key to institute table NOT NULL

column course\_name - varchar(100) NOT NULL

)

table student (

column student\_id - int - auto increment primary key NOT NULL

column institute\_id - int - foriegn key to institute table NOT NULL

column course\_id - int - foriegn key to course table NOT NULL

column student\_name - varchar(100) NOT NULL

column joining\_date - date NOT NULL

)

Notes: Always use HTTP Post and send any parameters as header parameters and JSON as POST body

Task 1: Create APIs for CRUD operations of institute, course and student tables

Task 2: Search API... take a search string/word as input .. this can be institute name or course name or student name , full or partial text search

fetch all the relevant records and give output response in the following format

[ { "institute\_name"; "dfgdgfd", "course\_name" : "asdssf", "student\_name": "wqrewer", "joining\_date" : "16 Sep 2024" },

{ "institute\_name"; "dfgdgfd", "course\_name" : "asdssf", "student\_name": "wqrewer", "joining\_date" : "16 Sep 2024" },

{ "institute\_name"; "dfgdgfd", "course\_name" : "asdssf", "student\_name": "wqrewer", "joining\_date" : "16 Sep 2024" } ]

Task 3: Create a stored procedure to convert student data to following report table and write an API to fetch the report details

i.e. how many students registered in a course monthwise based on joining date

table report\_student\_enrollment (

column year - int

column month - int

column institute\_name - varchar

column course\_name - varchar

column registered\_student\_count - int

)

API output should be like

[ { "Year" : 2024, "Month" : 08, "institute\_name"; "dfgdgfd", "course\_name" : "asdssf", "student\_count": 10},

{ "Year" : 2024, "Month" : 07, "institute\_name"; "dfgdgfd", "course\_name" : "asdssf", "student\_count": 7},

{ "Year" : 2024, "Month" : 06, "institute\_name"; "dfgdgfd", "course\_name" : "asdssf", "student\_count": 15},

{ "Year" : 2024, "Month" : 05, "institute\_name"; "dfgdgfd", "course\_name" : "asdssf", "student\_count": 2}]

Task 4: Create an API to upload student photo ... API parameters are student\_id and file uploaded

and save this to a location on server

Task 5: Create an API to download Student ID card ... API input parameters are student\_id

This should generate and download a pdf file that contains the details --> student photo, student name, course name, institute name and joining date

in a rectangular id card format convenient to you. No need for any beautification and complex formatting.