Project Development Phase(Sprint 3)

Team ID	PNT2022TMID04674
Project Name	Skill/Job Recommender Application

Pages:

Current Question : 3/10   15 minutes left   Submit Test

Which of the following is immutable by default in Rust but can be made mutable?
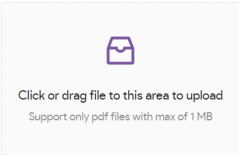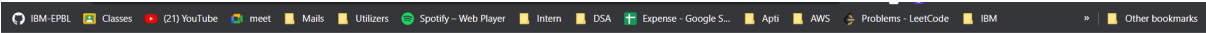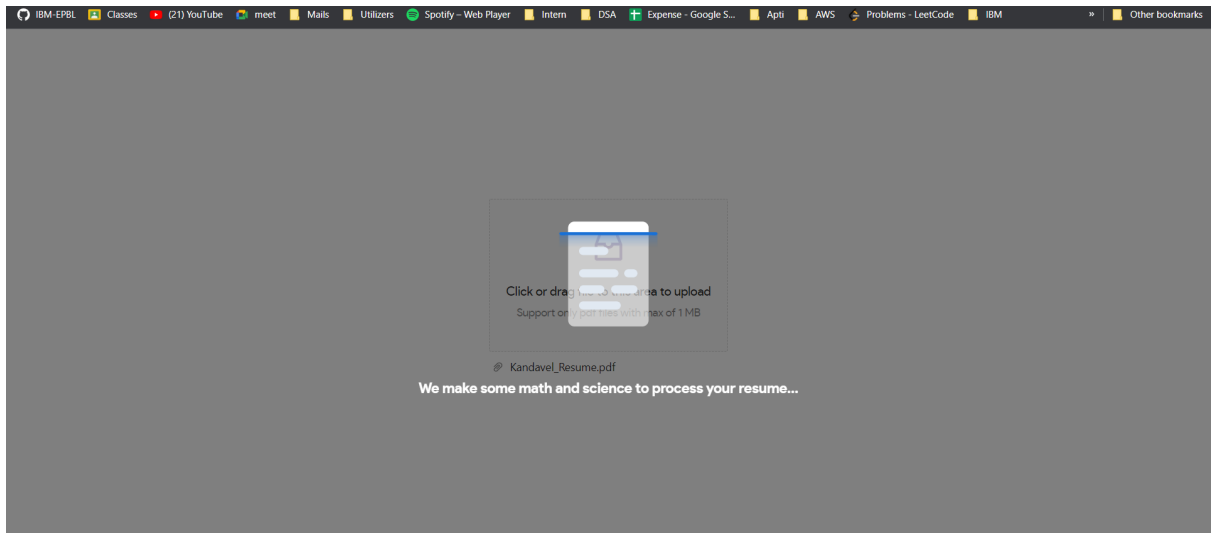
variable

const

user defined object

Depends on usage

Next   Saved answers : 3

Assessment Page

Click or drag file to this area to upload
Support only pdf files with max of 1 MB

Resume upload Page

Click or drag files to this area to upload

Support only pdf files with max of 1 MB

📎 Kandavel_Resume.pdf

We make some math and science to process your resume...

Resume uploading

Personal Details

Email (same as account)

sanjai@gmail.com

Name

Kongu Engineering College

Education

ECE | KEC | 9 | 🗑

📄 Add Another Education Field

Experience

1

Mobile Number

+919962857600

Details fetch from resume

Build pixel-perfect, robust and, accessible user interfaces on the web - Drive independently a project from inception to production deployment with a strong focus on performance and robustness - Follow our software development process i...

Read more

Recommended Assessments based on your skills

JS

Javascript

Total Questions : 10
Pass Percentage: 65%

Take Test

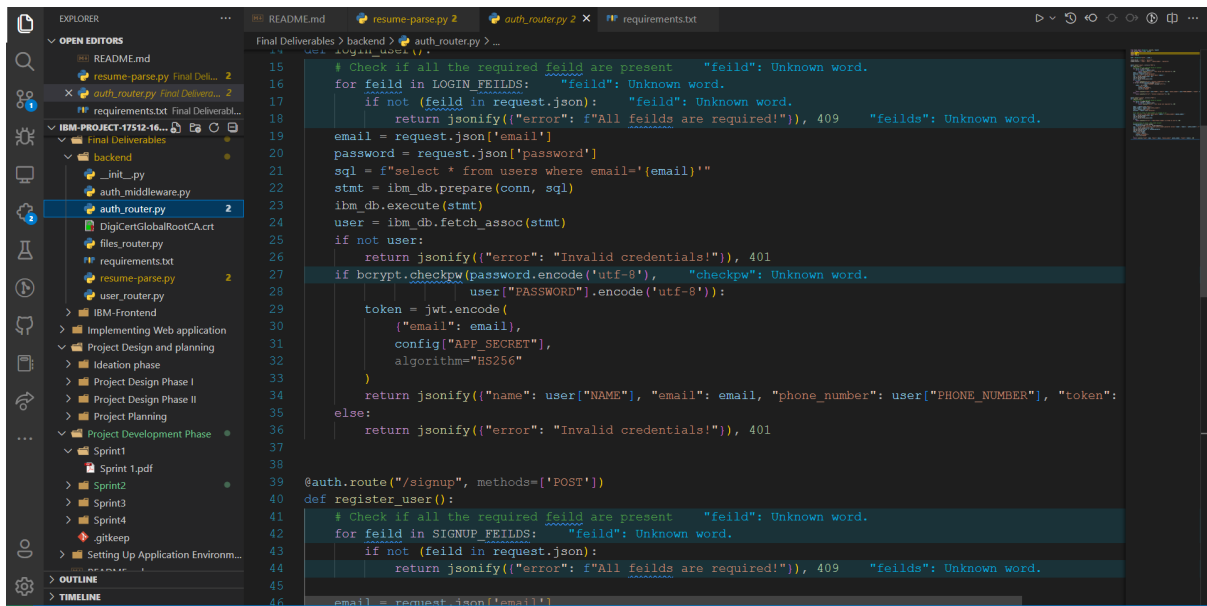**Are you wanna groom your skills? Here are some hackathons for you**
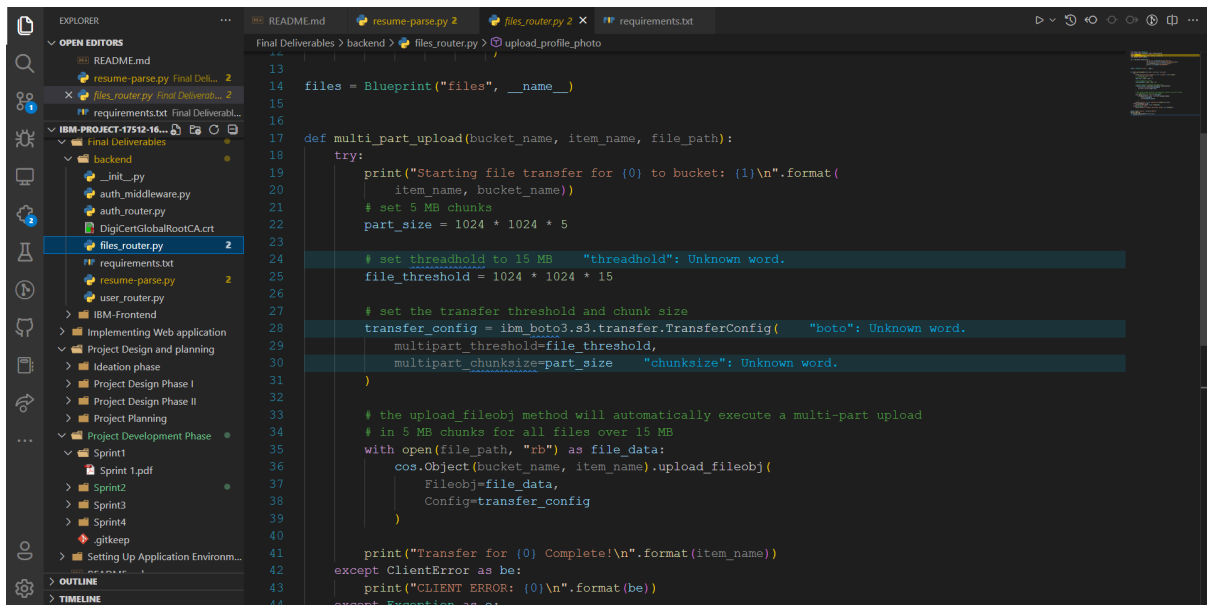
Assessment based on my Skillset

Code Examples:

```python
# Check if all the required feild are present
for feild in LOGIN_FEILDS:
    if not (feild in request.json):
        return jsonify({"error": f"All feilds are required!"}), 409
email = request.json['email']
password = request.json['password']
sql = f"select * from users where email='{email}'"
stmt = ibm_db.prepare(conn, sql)
ibm_db.execute(stmt)
user = ibm_db.fetch_assoc(stmt)
if not user:
    return jsonify({"error": "Invalid credentials!"}), 401
if bcrypt.checkpw(password.encode('utf-8'),
                  user["PASSWORD"].encode('utf-8')):
    token = jwt.encode(
        {"email": email},
        config["APP_SECRET"],
        algorithm="HS256"
    )
    return jsonify({"name": user["NAME"], "email": email, "phone_number": user["PHONE_NUMBER"], "token":
else:
    return jsonify({"error": "Invalid credentials!"}), 401


@auth.route("/signup", methods=['POST'])
def register_user():
    # Check if all the required feild are present
    for feild in SIGNUP_FEILDS:
        if not (feild in request.json):
            return jsonify({"error": f"All feilds are required!"}), 409
    email = request.json['email']
```

```python
files = Blueprint("files", __name__)


def multi_part_upload(bucket_name, item_name, file_path):
    try:
        print("Starting file transfer for {0} to bucket: {1}\n".format(
            item_name, bucket_name))
        # set 5 MB chunks
        part_size = 1024 * 1024 * 5

        # set threadhold to 15 MB
        file_threshold = 1024 * 1024 * 15

        # set the transfer threshold and chunk size
        transfer_config = ibm_boto3.s3.transfer.TransferConfig(
            multipart_threshold=file_threshold,
            multipart_chunksize=part_size
        )

        # the upload_fileobj method will automatically execute a multi-part upload
        # in 5 MB chunks for all files over 15 MB
        with open(file_path, "rb") as file_data:
            cos.Object(bucket_name, item_name).upload_fileobj(
                Fileobj=file_data,
                Config=transfer_config
            )

        print("Transfer for {0} Complete!\n".format(item_name))
    except ClientError as be:
        print("CLIENT ERROR: {0}\n".format(be))
    except Exception as e:
```

Full code - 🔗 **Github(Team id -PNT2022TMID04674)**