

## PYTHON API's

Q 1: Setup a FastAPI application in which following modules will be there with Postgres/mysql Connectivity.

### 1. Login Management:

#### a. User register API

```
router = APIRouter(
    prefix="/users",
    tags=['users']
)

get_db = database.get_db

@router.post(path='/register', response_model=schemas.User)
def register(request: schemas.User, db: Session = Depends(get_db)):
    existing_user = user.fetch(request.username, db)
    if existing_user :
        raise HTTPException(status_code=status.HTTP_400_BAD_REQUEST, detail = "Username is already registered.")

    new_user = user.create(request, db)
    if new_user :
        return new_user
    raise HTTPException(status_code=status.HTTP_500_INTERNAL_SERVER_ERROR, detail="Could not create user.")
```

#### b. Login API

```
from user_management.routers import hash

BEARER = "bearer"

router = APIRouter(tags=['Authentication'])

get_db = database.get_db

@router.post('/login')
def login(request: OAuth2PasswordRequestForm = Depends(), db: Session = Depends(get_db)):
    user = db.query(models.User).filter(
        models.User.email == request.username).first()
    if not user:
        raise HTTPException(status_code=status.HTTP_404_NOT_FOUND,
                            detail=f"Invalid Credentials")
    if not Hash.verify(user.password, request.password):
        raise HTTPException(status_code=status.HTTP_404_NOT_FOUND,
                            detail=f"Incorrect password")

    access_token = token.create_access_token(data={"sub": user.email})
    return {"access_token": access_token, "token_type": BEARER}
```