

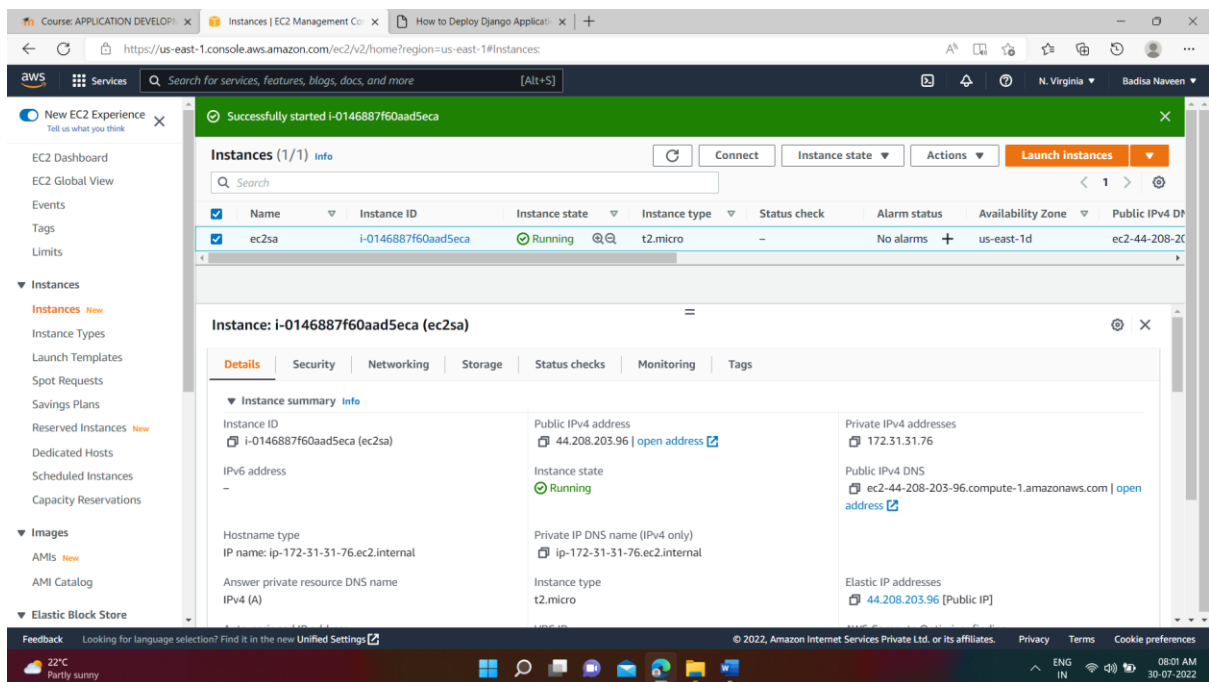
20CS3234AA – ADC

Name: Badisa Naveen

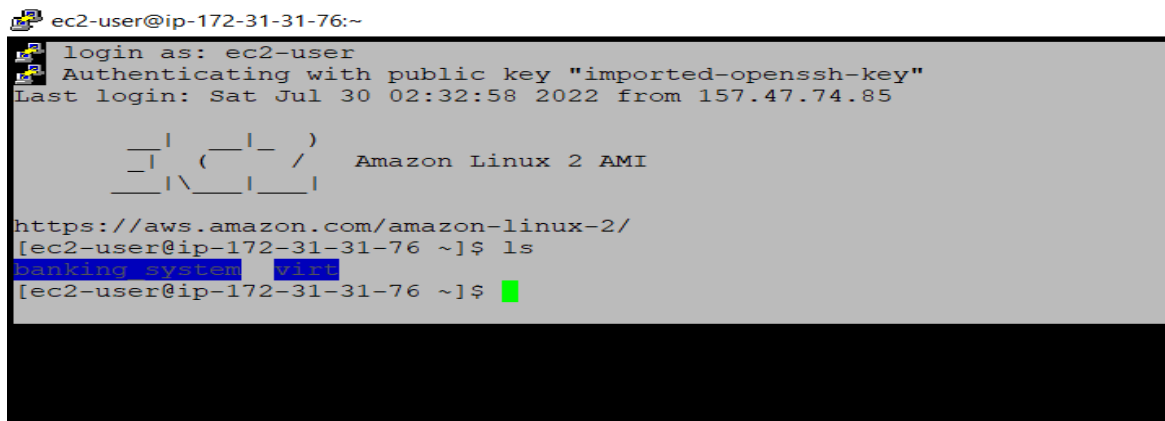
Reg.no: 2000031509

1. An order for banking web application is been received by your IT company. They decided to go for it. They decided the database as RDS, since banking operations will have transactions and RDS supports transactions well. So, your company need to test the skill of employees before assigning this project. The higherups wants you to create a simple application of your choice and connect to MySQL database in cloud and do the CRUD operations. Record the process as step by step screen shots for later audit by your higherups.

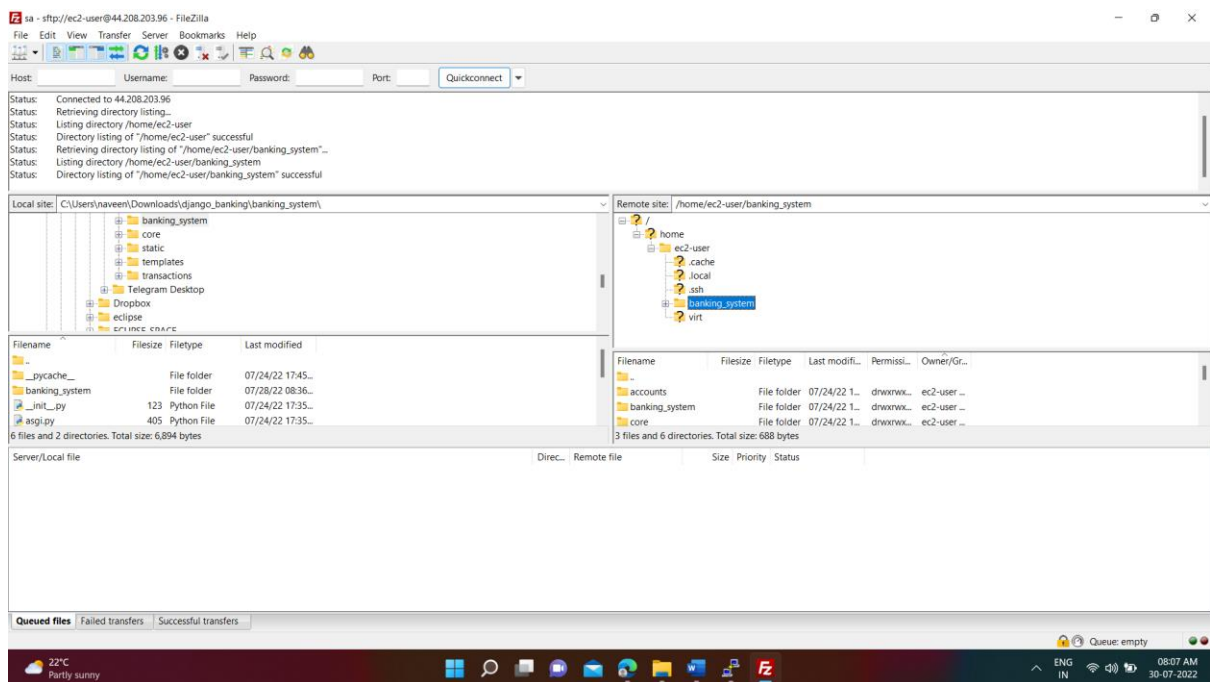
Steps1: Create an ec2 with elastic IP address



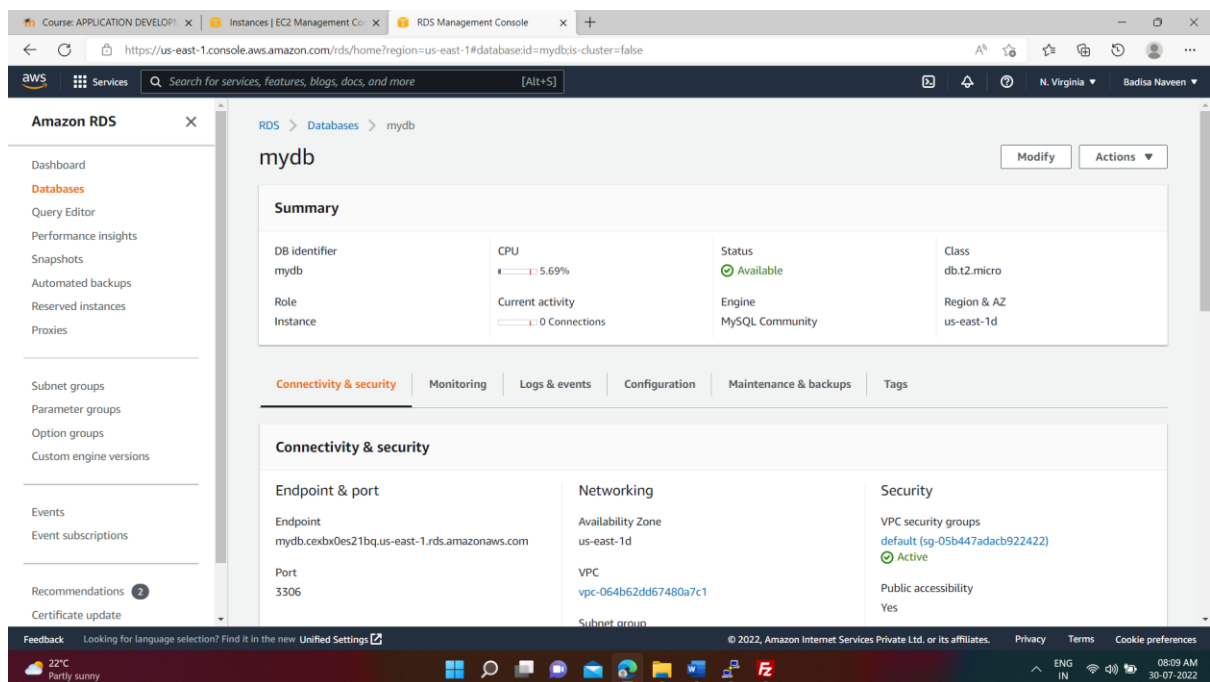
Step2: Connect ec2 through putty



Step3: Transfer files from local device to ec2 through filezilla



Step4: Create a RDS Wait until endpoint generates



Step5: Connect MYSQL RDS to Django application

```
ec2-user@ip-172-31-31-76:~/banking_system/banking_system
GNU nano 2.9.8 settings.py

'DIRS': [BASE_DIR / 'templates'],
'APP_DIRS': True,
'OPTIONS': {
    'context_processors': [
        'django.template.context_processors.debug',
        'django.template.context_processors.request',
        'django.contrib.auth.context_processors.auth',
        'django.contrib.messages.context_processors.messages',
    ],
},
},
}

WSGI_APPLICATION = 'banking_system.wsgi.application'

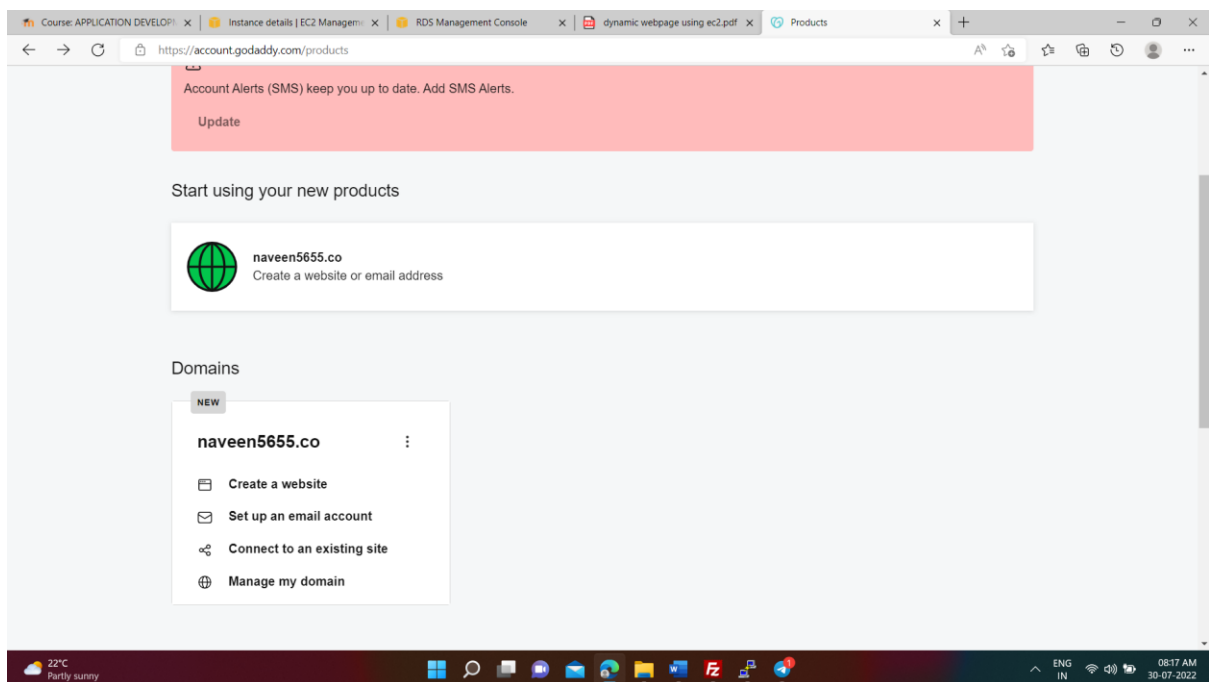
# Database
# https://docs.djangoproject.com/en/3.1/ref/settings/#databases

DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.sqlite3',
        'NAME': BASE_DIR / 'db.sqlite3',
        'ENGINE': 'django.db.backends.mysql',
        'HOST': 'mydb.cxibx0es2lbg.us-east-1.rds.amazonaws.com',
        'USER': 'root',
        'PASSWORD': 'postgres',
        'NAME': 'mydb',
    }
}

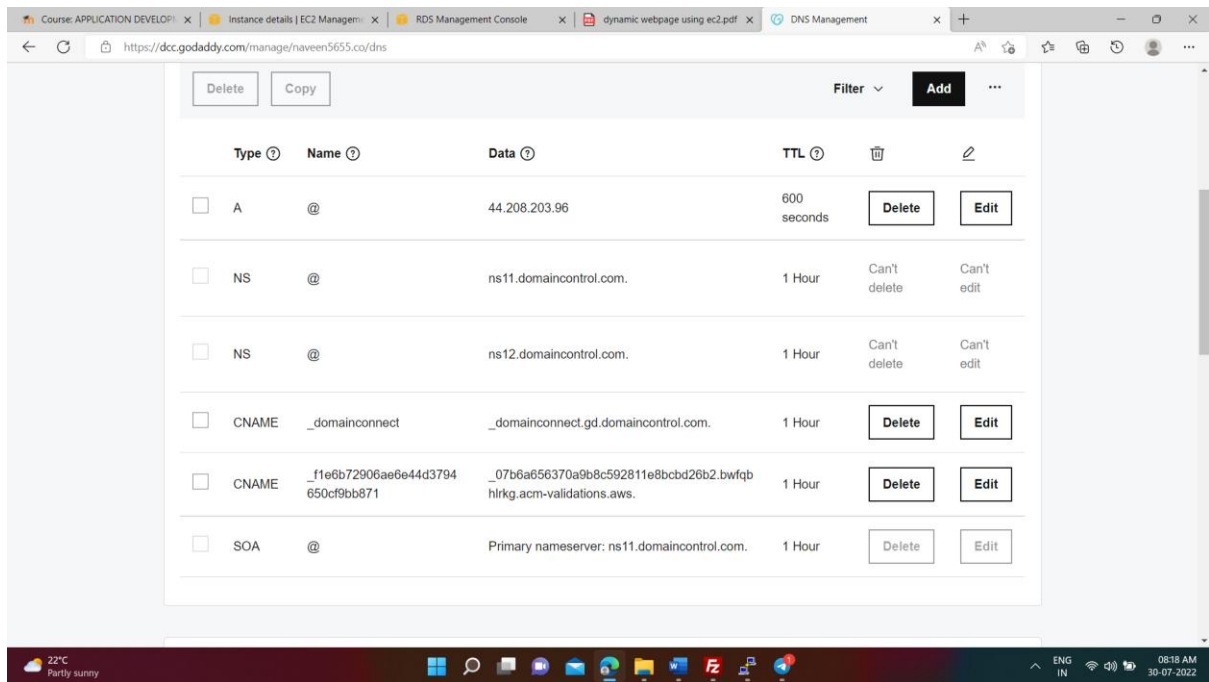
# Password validation
# https://docs.djangoproject.com/en/3.1/ref/settings/#auth-password-validators

AUTH_PASSWORD_VALIDATORS = [
    {
        'NAME': 'django.contrib.auth.password_validation.UserAttributeSimilarityValidator',
    },
    {
        'NAME': 'django.contrib.auth.password_validation.MinimumLengthValidator',
    },
    {
        'NAME': 'django.contrib.auth.password_validation.CommonPasswordValidator',
    },
]
```

Step6: Open godaddy and create one domain



Step7: change name servers and connect to ip address of ec2

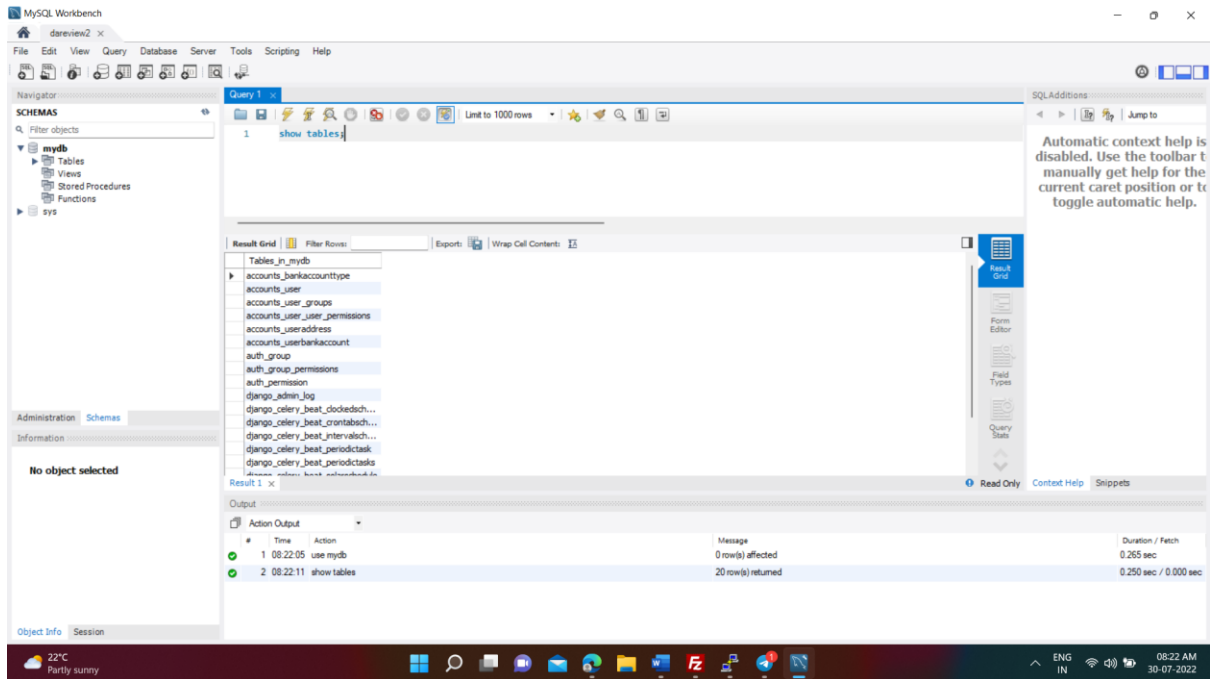


Step8: Run the server

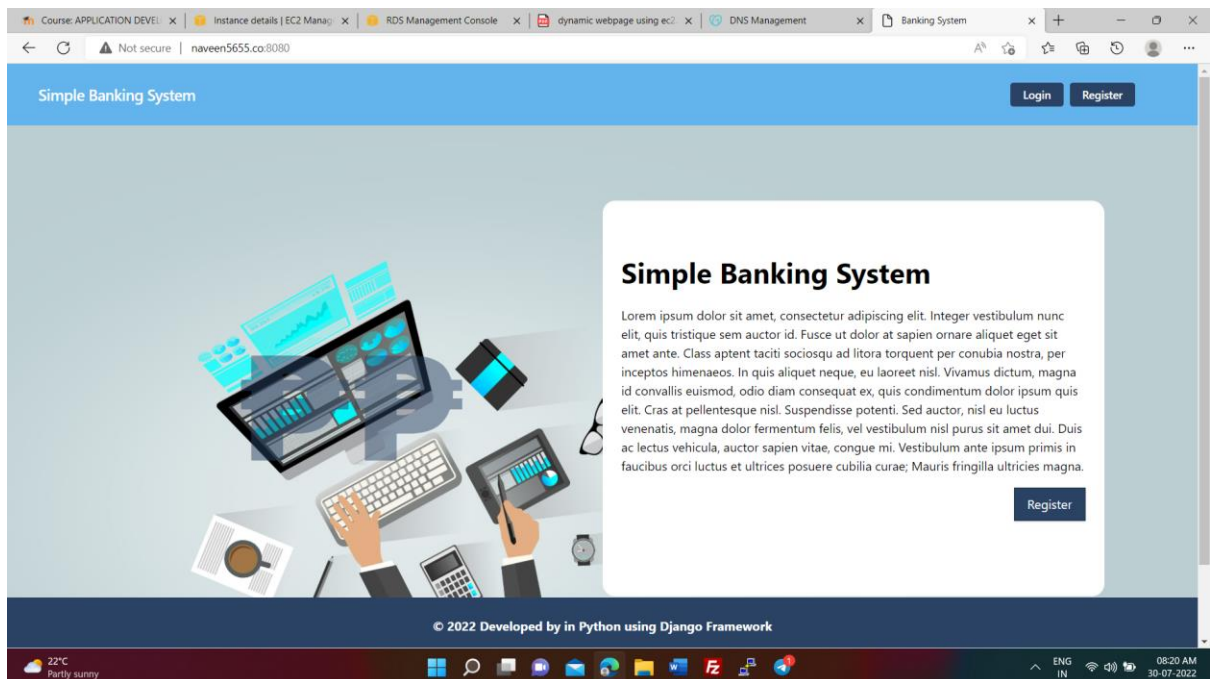
```
(virt) [ec2-user@ip-172-31-31-76 banking_system]# cd ..
(virt) [ec2-user@ip-172-31-31-76 banking_system]$ ls
accounts  banking_system  core  requirements.txt  static  templates  transactions
(virt) [ec2-user@ip-172-31-31-76 banking_system]$ ./runserver.py
Watching for file changes with StatReloader
Performing system checks...

System check identified no issues (0 silenced).
July 30, 2022 - 10:49:14
Django version 3.1.8, using settings 'banking_system.settings'
Starting development server at http://0.0.0.0:8080/
Quit the server with CONTROL-C.
```

Step9: Connect mysql in workbench



Step9: type the Domain name in the Browser



Step10: click on register

Simple Banking System

Register

FIRST NAME: Badisa

LAST NAME: Naveen

EMAIL: 2000031509@kluniversity.in

ACCOUNT TYPE: Savings

GENDER: Male

BIRTH DATE: dd-mm-yyyy

PASSWORD: *****

PASSWORD CONFIRMATION: *****

STREET ADDRESS: singannagudem

CITY: Vijayawada

© 2022 Developed by in Python using Django Framework

Step11: Register successfully

Simple Banking System

Home Deposit Withdraw

Welcome, Badisa Naveen! Logout

Success!
Thank You For Creating A Bank Account. Your Account Number is 100005663.

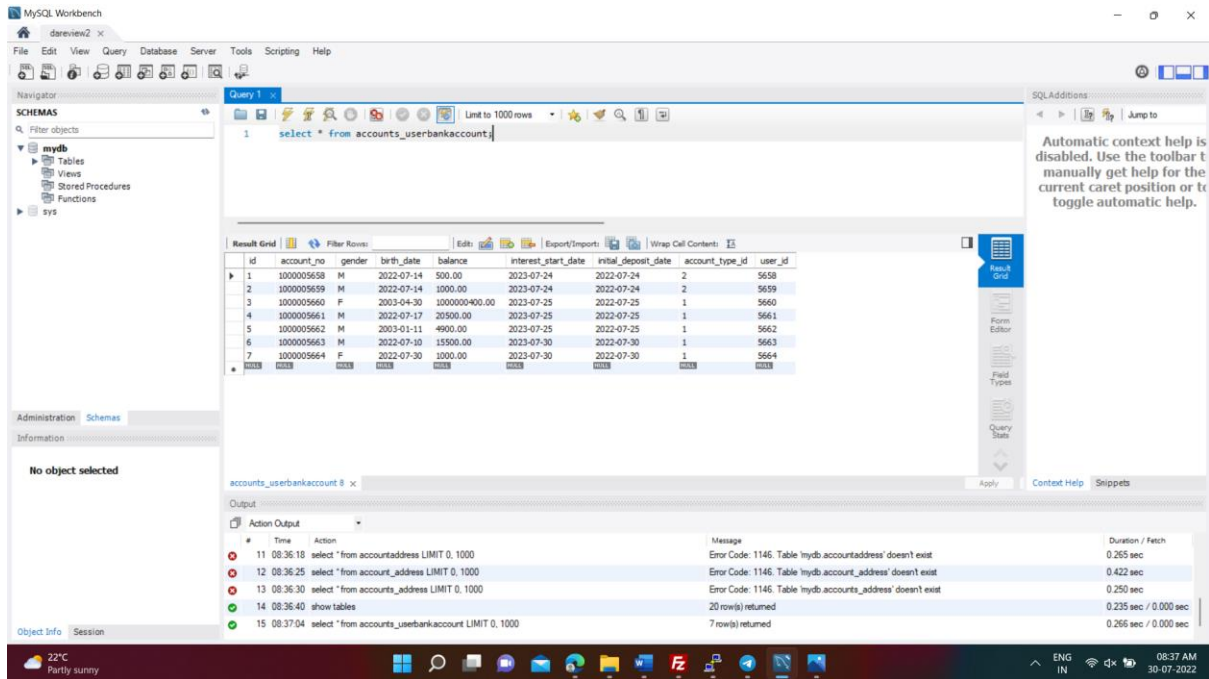
Deposit Money to Your Account

Amount: Amount

Submit

© 2022 Developed by in Python using Django Framework

Step12: Check register data in MYSQL workbench



Step13: add amount

