**Pay APP – Report**

The Pay APP is a web-based application for the transactions of payment. Users can send or receive payments from registered users or request payments from registered users. The application has been developed using the MVC architecture in Django that has different layers that are following.

* Presentation Layer
* Business Logic Layer
* Data Access Layer
* Security Layer
* Web Service Layer

## Presentation Layers

The presentation layer is responsible to display an attractive view to the end user for the related task. In the Pay APP application, firstly users were registered and signed with different GUIs and then use the app for sending and receiving payments. To complete the main objectives of the application, the main two apps names as ‘register’ and ‘payapp’ were developed that provide the designes for the following functionalities

Register APP

* Register a new user.
* Log-in user.

Pay-App

* Transfer Payment to Register User
* Request Payment from Register User
* Get a Notification upon Receiving the Request
* View Notification
* Accept\Reject Request
* View Transaction History

There were a few GUIs that design for the admin users only.

* View all Users.
* View transactions of all users
* Register new user as admin

## Business Logic Layer

In the business logic layer, the methodology for serving the designed templates was implemented. The business logic layer implements the functionality to facilitate the registered user and admin user in the following ways.

**Register User**

By using the business logic layer, a register user can do the following task.

* Register himself in the application.
* Login in the application
* Make direct payments to other registered users.
* Request payments from registered users
* View all their transactions.
* View their Profile Info and Balance Information

While the administrator user can avail the following facilities from the business logic layer:

* Can get all user info.
* Can get the history of all transactions in the application.
* Register more users as administrator.

## Data Access Layer

In the data access layer, a few models were implemented that are responsible for making schema in the MySQL db. By following the development of schema these models are also responsible for storing and retrieving of the data. The detail of each model is following.

A User table is already available in the Django application that has the following fields:

* Username
* Email
* Password
* First Name and Last Name (Optional Fields)

A User Extension model was developed that fulfills the requirements of the user by implementing the following fields:

* First Name and Last Name (Compulsory)
* Balance
* Currency (USD, EUR, GBP)

The user extension model was also linked with the built-in User model with a one-on-one relationship. Furthermore, a Transaction History model was developed that linked with the User model to store the user transactions. By following the implementation of the Data access layer application can perform the following actions:

* Store the user data.
* Retrieve the User data.
* Store the Transaction History (Transfer/Request)
* Retrieve Transaction History

## Security Layer

The security layer of the application ensure that the validated users are using the application. Moreover, the security of data, privileges and authorities is also ensuring by the security layer. In the security layer of the pay app, we ensure the following security measures:

* Authentication Functionality: In the authentication functionality, a user was authenticated with a valid username and password for using the application. Moreover, the authentication of the user was checked every time when he tries to access the application functionality.
* Access Control Navigation Functionality: Before providing the functionality to the user, the role or privileges of the users were also ensured. Some functionalities are restricted to the admin users only and must be provided to that user only.
* HTTPS Functionality: Not implemented.
* Web Security: Every time when a user tries to access the data access layer for storing data, the CSRF security was also validated with a CSRF token in Django.
* Initial administration Registration: In the user extension table a record was entered to create an administrator. When the migration runs, the migration will create an admin user after the development of the user model.

## Web Service

The Web Service is the requirement of the application to implement the functionality of the currency exchange. But the separate web service was not implemented in the application. While the currency exchange functionality was implemented in py file inside the register app module.

# **User Manual**

The user manual was prepared by the video demo that is attached with this file in video format.