**Django Webapps Report**

**This report includes following sections:**

1. **Presentation**: Through this layer users can interact with each other and administrators.

Following functionality is implemented:

* User can view all their transactions, make direct payments to other registered users, request payments from registered users and accept, decline requests of other users.
* Administrators can see user accounts and all payment transactions and register new admins.

1. **Business Logic Layer**: Through this layer we have implemented the logic layer consisting of views.

Following functionality is implemented:

* User can view all their transactions, make direct payments to other registered users, request payments from registered users and accept, decline requests of other users.
* Administrators can see user accounts, view all accounts and user balances, view all payment transactions, and register new admins.
* Internal structure given for project is also followed while implementing logic.
* Project is named as webapps2023.

database:  db.webapps  
context path:  /webapps2023

1. **Data Access Layer**: Through this layer we can save all the data in appropriate model related to views.

Following functionality is implemented:

* SQLITE is used as Relational Database Management System (RDBMS).
* Model contains all the essential fields and information related to data and can be viewed by admins in admin portal, we use python database access API for accessing objects.
* Database is named as db.webapps.

1. **Security Layer**: This is a multi-user web application which can be seen in video and screenshot given with this report.

Following functionalities are implemented:

* Administrators access their own page.
* Users cannot access functionality of administrators.
* User can only see their own page and not other users page or information.
* Users can register, login, log out from web application.
* Access control to restrict access to web pages to non-authorised users.
* Cross-site scripting (XSS), Cross-site request forgery (CSRF), SQL injection, and Clickjacking protection.
* Https has been enabled on website.
* Password is set as given in the assignment report.

1. **Web Services**: Through this layer we can access a get API. It is a REST service accessed by business logic layer.

Following functionality is implemented:

* currency conversion RESTful web service that responds only to GET requests has been implemented.

which has URL as follows:

**baseURL/conversion/{currency1}/{currency2}/{amount\_of\_currency1}**

* The RESTful web service returns an HTTP response with the conversion rate (currency1 to currency2) or the appropriate HTTP status code if one or both provided currencies are not supported.