|  |  |
| --- | --- |
| * **Presentation Layer (20%)** | * 10% - Full marks will be given if all required templates are written and correctly connected with view functions in a way that makes sense even if no other functionality is implemented at the business logic and data access layers. The set of correctly implemented templates includes .html pages required to perform security-related actions. * 5% - Full marks will be given if all required conversions and validations are done. This highly depends on the way you design your pages. In most cases, standard validations and conversions should be enough. Full marks will be given to assignments that support full and correct page navigation by explicitly specifying navigation rules in URLConf modules. * 5% - The appearance of web pages will be marked. It is advisable to use crispy forms and bootstrap. |
| * Users can view all their transactions |  |
| * Users can make direct payments to other registered users |  |
| * Users can Request payments from registered users |  |
| * Admins should be able to see all user accounts |  |
| * Admins should be able to see all payment transactions |  |
| Admins can register new administrators |  |
| 4.2. The Business Logic Layer (20%) | Full marks will be given if all required business logic is implemented in a set of views, which must include appropriate annotations for supporting transactions if and when required. |
| * Users should be able to View all their transactions |  |
| * Users should be able to Make direct payments to other registered users |  |
| * Users should be able to Request payments from registered users |  |
| * Admins should be able to View all user accounts and balances |  |
| * Admins should be able to View all payment transactions |  |
| * Admins should be able to Register more administrators |  |
| 4.3. The Data Access Layer (10%) | 15% Full marks will be given if all access to application data is handled through models and properly relating tables to each other. |
| * Have SQLite |  |
| 4.4. The Security Layer (20%) |  |
| 5% - Authentication functionality   * Full marks will be given if users can register, login and logout. An admin must be registered in the system upon activating the models |  |
| * 4% - Access control when navigating through templates - Access to templates must be restricted to authorised actors (users and admin). |  |
| * 5% - HTTPS functionality - Communication with the web application using HTTP |  |
| 5% - Web security protection functionality - Protection against cross-site scripting (XSS), cross-site request forgery (CSRF), SQL injection, and clickjacking attacks needs to be in place |  |
| * 1% - Initial administration registration - Upon activating the models, a single administrator account (username: **admin1**, password: **admin1**) must be present. Only an administrator can register more administrators through the restricted admin pages |  |
| 4.5. Web Services (10%) |  |
| A currency conversion RESTful web service that responds only to GET requests. The exported resource should be named conversion in a path such as the following:  baseURL/conversion/{currency1}/{currency2}/{amount\_of\_currency1}  The RESTful web service should return an HTTP response with the conversion rate (currency1 to currency2) or the appropriate HTTP status code if one or both of the provided currencies are not supported. |  |
| 4.6. RPC with Apache Thrift (10%) All transactions must be timestamped by accessing a 'remote' Thrift timestamp service (which is deployed on the same server as your system). When requested, the service should return the current date and time to your system. The Thrift server can be implemented as a Django application, which uses a separate thread to accept time-stamping requests at port 10000 (or any other available port). |  |
| **4.7 Deployment on Amazon AWS (5%)** You must successfully deploy and run the application on an Amazon EC2 virtual machine to get 5%. To get full marks, you must submit screenshots of the commands issued on the console to run your Django web application and screenshots of the application running on the cloud where the URI of the application is shown. To verify that you have indeed deployed the application, the tutor may ask you, during the marking period, to run the server and deploy your application for it to be tested. You have already been invited to join AWS Academy and should be able to access the required AWS resources. Sussex is an active AWS Academy member, meaning you will be given some free credits to use AWS. These credits will be enough to deploy this project. Some tutorials on using AWS have been provided in the module lab classes. |  |
| 4.8 Report (5%)  he report must not have more than **16 pages**.   * 2% - The report should include seven sections: Presentation Layer, Business Logic Layer,  Data Access Layer, Security Layer, Web Services, RPC, and Cloud. In each section, you need to specify what has been fully or partially implemented or if any requirement has not been implemented. * 3% - The report should include a user manual to assist users in using your application. This can be one of these two options:   + a list of screenshots of running your project. You need to show all the working parts, e.g., the main interface, the user/admin registration, request payment, payment transaction,  REST service, etc. These screenshots should show all the user interfaces and any related data in the database.   + or a short walkthrough video/demo that shows all the working parts. |  |