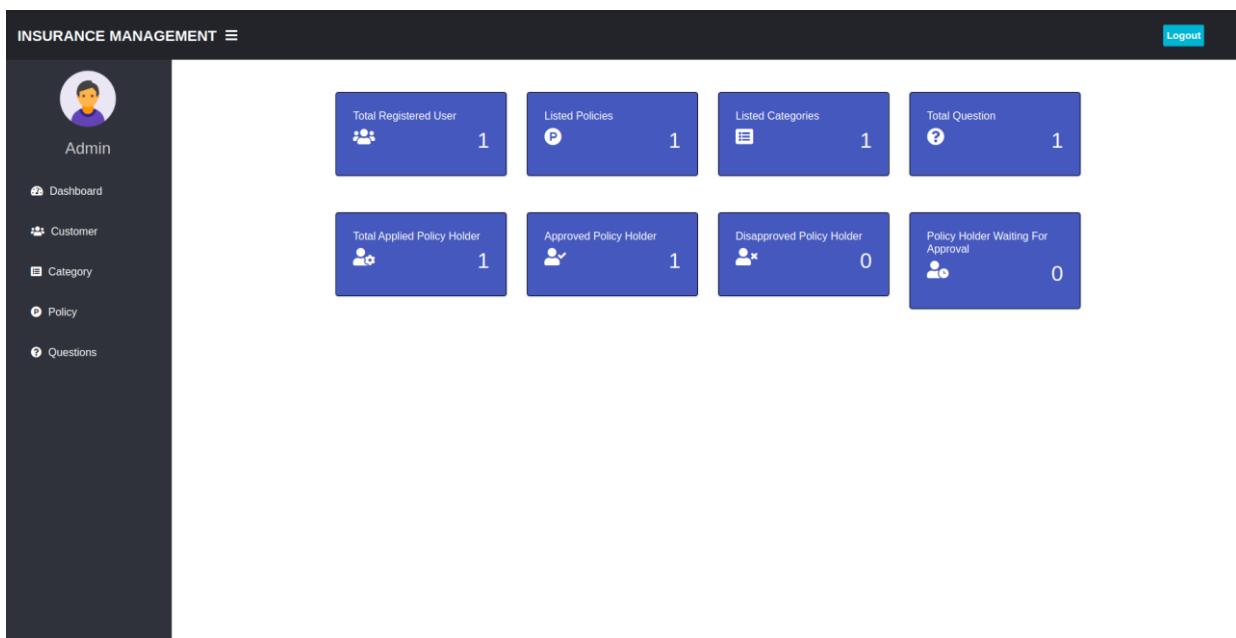


Django Framework

Python

Assessment Test

- Write a program to demonstrate the Insurance Management application using of Django web framework
- Prepare demonstration of Python Insurance Management application under software development principles and follow coding protocols
- The project was built in a MVT concept create an effective interactive interface.
- In this application Perform all operations using ORM techniques



- Make sure each business logic is denoted with appropriate comments and make your code interactive and represent clean and clear output on browser.
- In this app admin can view all register users , policy holder , approve and reject policy holder's request from the dashboard
- Include base.html file which contain header and sidebar portion only. Add it in all container page
- Use models.py file which contains all the tables related to this application

- **Store all data in MySQL database.**

The screenshot shows a registration form titled "Register". At the top left is the application name "MyDjangoApp". At the top right is a "Register" button. The form itself has a light gray header with the word "Register". Below this are seven input fields: "Username*", "Email*", "Password*", "Confirm password", "First Name", "Last Name", and "Phone Number". Each field is accompanied by a label and a text input box. At the bottom right of the form is a green "Register" button.

- **Make sure above registration page contain validation using of javascript – store above data using of ajax module.**
- **After Customer registration send confirmation mail to register email address.**
- **Email credential and password must store in settings.py and global use it in entire the application**
- **Apply forgot password feature in this application**

The screenshot shows a "Forgot Password?" form. At the top center is the title "Forgot Password?". Below it is a label "Email" followed by a text input field with placeholder text "Enter email". At the bottom of the input field is a dark blue "Submit" button. Below the input field is a horizontal line. Underneath the line, there are two links: "Create A New Account" and "Back To Login", both in blue text.

- **Create separate utils.py for unique business logics**
- **Manage proper naming conversion – create appropriate objects name.**

- **create separate file for all business logics and make them reusable - use modules concepts for implements above logic**
- **Make sure validation proper given – each options - display appropriate message if user enter invalid input**
- **Make sure code prevent from unexpected exception return to the previous screen and accept all details again.**
- **Make sure Database normalize manage in this project work**
- **After execution of each option confirmation message must be displayed.**
- **Remove option must ask to user for question id to delete and again ask for confirmation (Y/N) before deletion and display proper message after deletion**
- **Program should not be terminated till the user Exit it**
- **Developer needs to test his product before launching it into the market**
- **After completion this project upload it on GitHub**
 - **Upload all features in develop branch after completion all features merge it with main branch**

