

Protegrity Coding Test Documentation

Submitted by **Akshay Balaso Karande**

For Position of Full Stack Developer (Python)

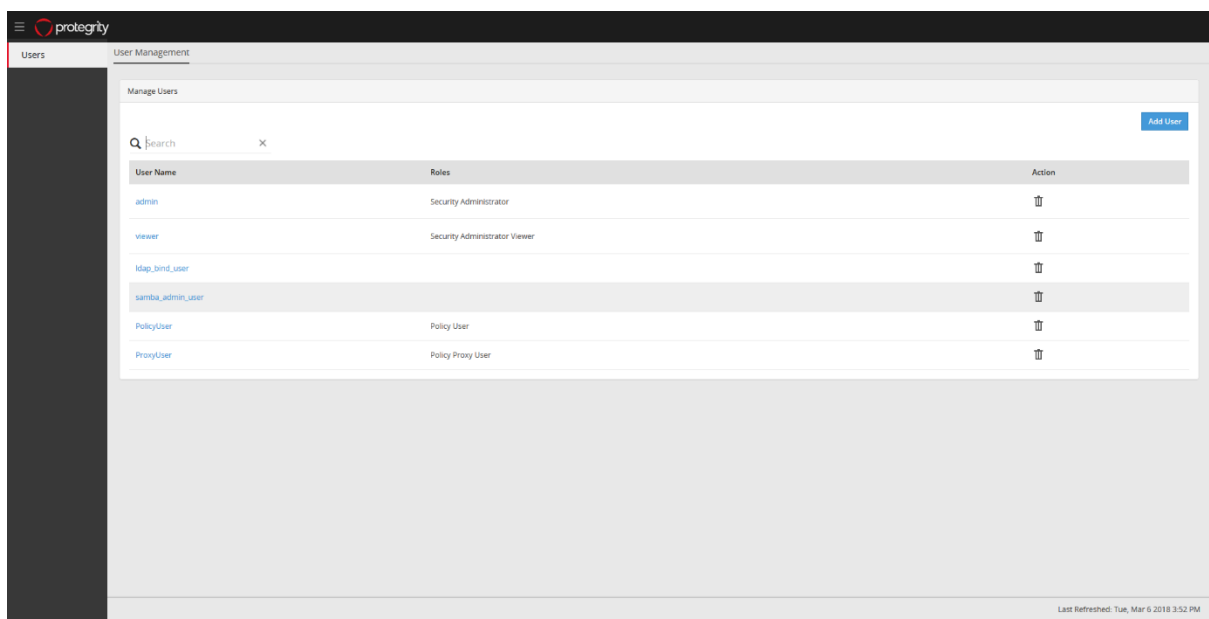
Here, before im going to submit my coding task . I want to make it clear about question and his desired input with required output

So im attaching your assignment questions here.

Assignment

UI Requiements:

Page 1: User Management



Page 2 Add User With Roles

The screenshot shows the 'New User' form in the Prolegity User Management interface. The form is divided into two main sections: 'General Info' and 'Roles'.

General Info:

- First Name:
- Last Name:
- User Name:
- Description:

Roles:

Select Role(s) from the list below 1 selected

<input type="checkbox"/>	Role Name	Description	roles
<input type="checkbox"/>	Security Administrator Viewer	Security Administrator Viewer Role	Cloud Gateway Viewer, Security Viewer, Appliance CLI Viewer, Appliance web viewer
<input type="checkbox"/>	Shell Accounts	Accounts that have shell access	Shell (non-CLI) Access
<input checked="" type="checkbox"/>	Security Administrator	Security Administrator Role	Security Officer, Appliance web manager, Appliance CLI Administrator, Export Certificates, Cloud Gateway Admin, Cloud Gateway Auth, DPS Admin, Directory Manager, Export Keys, RLP Manager
<input type="checkbox"/>	Directory Administrator	Local LDAP Administrator Role	Directory Manager

Cancel Add User

Last Refreshed: Tue, Mar 6 2018 3:52 PM

Page 3 Update/edit User:

The screenshot shows the 'Update/edit User' form in the Prolegity User Management interface for the user 'admin'. The form is divided into two main sections: 'General Info' and 'Roles'.

General Info:

- First Name: admin
- Last Name: admin
- User Name: admin
- Description:

Roles:

Select Role(s) from the list below 1 selected

<input type="checkbox"/>	Role Name	Description	roles
<input type="checkbox"/>	Security Administrator Viewer	Security Administrator Viewer Role	Cloud Gateway Viewer, Security Viewer, Appliance CLI Viewer, Appliance web viewer
<input type="checkbox"/>	Shell Accounts	Accounts that have shell access	Shell (non-CLI) Access
<input checked="" type="checkbox"/>	Security Administrator	Security Administrator Role	Security Officer, Appliance web manager, Appliance CLI Administrator, Export Certificates, Cloud Gateway Admin, Cloud Gateway Auth, DPS Admin, Directory Manager, Export Keys, RLP Manager
<input type="checkbox"/>	Directory Administrator	Local LDAP Administrator Role	Directory Manager

Save

Last Refreshed: Tue, Mar 6 2018 3:52 PM

This are the UI requirement's

Functional Requiements:

now we will see the functional requirement given by you.

Create a user management page that should function as the following.

- 1) The page should look like 01-UserManagement-main_page.png. If the user will click on the hamburger icon on the top left, it will close the menu
- 2) The page will list all the users that has been already created.
- 3) User can click on the trash can icon to delete an existing user.
- 4) User can click on Add User button create a new user which should bring the new user screen.
- 5) clicking user name in the list will bring the edit user screen where user information and roles can be changed and the changes can be saved,

Create the page and the rest API's that can be used to implement the functionality. The user information, roles, etc can be saved in json files, SQLite database or any other database that the candidate is comfortable with.

Solution

Software requirement:

1- Python 3+

2- MySQL 5

UI

Page 1: User Management

Close x

Protegrity

User

Add User

User Management

Manage User

Q Search..

ADD USER

USER NAME	ROLES	ACTION
SagarTodkar	abc	
sujit	developer	
observer	abc	
shormal	abc	

Page 2:Add User

Close x

Protegrity

User

Add User

User Management

New User

CANCEL

ADD USER

1 - General Info

First Name

First

Last Name *

Last

User Name *

Username

Description

Description

Roles

select role(s) from the list bellow

Selected

SELECT	ROLE NAME	DESCRIPTION	ROLE
<input type="checkbox"/>	abc	hi im new here so my role is observing everyone and do my work	observer
<input type="checkbox"/>	developer	software developer	fresher
<input type="checkbox"/>	Marketing specialist	he is doing Marketing of our product	Marketing
<input type="checkbox"/>	Analyst	he does the anlysis of our client and his	Business analyst.
<input type="checkbox"/>	Accountant	Manage every transactions	Accountant

Page 3: Update/Edit User:

The screenshot shows a web application titled 'Protegrity' with a 'User Management' section. On the left is a dark sidebar with 'Close x', 'User', and 'Add User' options. The main content area has a header 'User Management' and a sub-header '< Back Admin' with a 'SAVE' button and a trash icon. Below this are two sections: 'General Info' and 'Roles'.

General Info

First Name	sagar
Last Name *:	Todkar
User Name *:	SagarTodkar
Description	<input type="text" value="Description"/>

Roles

select role(s) from the list below **Selected**

SELECT	ROLE NAME	DESCRIPTION	ROLE
<input type="checkbox"/>	abc	hi im new here so my role is observing everyone and do my work	observer
<input type="checkbox"/>	devloper	software devloper	fresher
<input type="checkbox"/>	Marketing specialist	he is doing Marketing of our product	Marketing
<input type="checkbox"/>	Analyst	he does the anlysis of our client and his	Business analyst.

How Functional Requirement's implemented:

On the basis of UI screen which is given by you I created a MySQL database with three table One is for "roles", second is for "user" and third one is "user_roles".

Here is a syntax of creating database and tables:

```
create database protegrity;  
use protegrity;
```

Table 1:

This table consist role_id as primary key, role name and description of role.

```
create table roles(  
role_id integer primary key AUTO_INCREMENT,  
role_name varchar(100),  
description varchar(300),  
roles varchar(300)  
);
```

Table 2:

This table consist user_id as primary key, first name , last name , user name of user and also description of user.

```
create table user(  
user_id integer primary key AUTO_INCREMENT,  
first_name varchar(30),  
last_name varchar(30),  
user_name varchar(30),  
description varchar(300)  
);
```

Table 3: This table is created to maintain a relationship between user and roles (many to many relationship in between both the tables that's way third table is formed)

```
create table user_roles  
(  
user_role_id integer primary key AUTO_INCREMENT,  
role_id integer,  
user_id integer,  
FOREIGN KEY (role_id) REFERENCES roles(role_id),  
FOREIGN KEY (user_id) REFERENCES user(user_id)  
);
```

Inserting Roles in role table

```
INSERT INTO `roles` VALUES
```

```
(1,'abc','hi im new here so my role is observing everyone and do my work','observer'),  
(2,'devloper','software devloper','fresher'),  
(3,'Marketing specialist','he is doing Marketing of our product ','Marketing '),  
(4,'Analyst','he does the anlysis of our client and his','Business analyst. '),  
(5,'Accountant','Manage every transactions ','Accountant');
```

Inserting data into user table:

```
INSERT INTO `user` VALUES (1,'abc','xyz','observer',''),  
(2,'shankar','Tormal','shormal',''),  
(3,'swapnil','khatvkar','swappy','java developer'),  
(4,'sujit','gaikwad','sujit',''),  
(5,'','sdfg','dsfg',''),  
(6,'','sdfg','dsfg',''),  
(7,'sagar','Todkar','SagarTodkar','reliance');
```

Inserting data into user_roles table:

```
INSERT INTO `user_roles` VALUES (2,1,2),(4,1,1),(5,2,4),(6,1,7);
```

Basically the third table data is inserted when you linked user with roles.

To Run this project in your environment you have to

Step 1: install above software's which is mentioned in software requirement

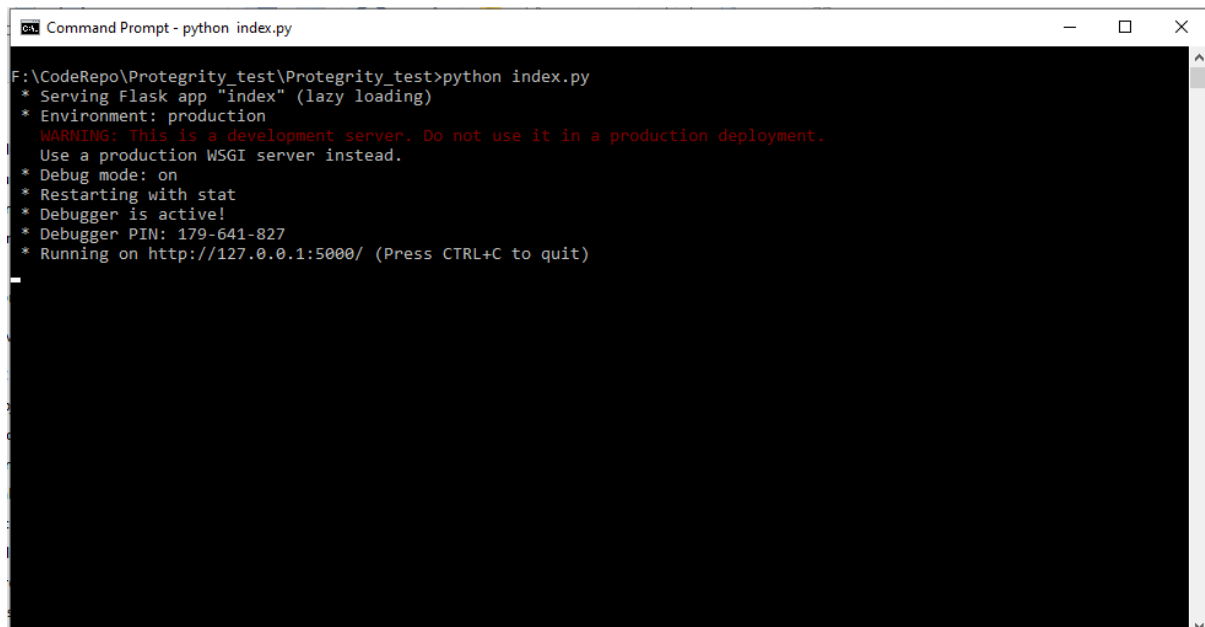
Step 2: Create database

Step 3: Install pip in your machine for that just refer google.

Step 4: Then open your command prompt navigate it to project directory then run following command "pip install requirement.txt".

Step 5: Then open project directory from file manager go to the project directory, open config folder then you can see there is file with name of "config.yml", Open that file in notepad or any other text editor just replace you database properties from that page on specific variable.

Step 6: on the same directory "python index.py" run this command.



```
Command Prompt - python index.py
F:\CodeRepo\Protegrity_test\Protegrity_test>python index.py
* Serving Flask app "index" (lazy loading)
* Environment: production
  WARNING: This is a development server. Do not use it in a production deployment.
  Use a production WSGI server instead.
* Debug mode: on
* Restarting with stat
* Debugger is active!
* Debugger PIN: 179-641-827
* Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
```

Command prompt shows like that , then just open your browser and enter the link which is given in command prompt after running index.py

<http://127.0.0.1:5000/>

where 5000 is default port number you can change this port to 5001

I also mentioned that how to change port in index.py at line number 243.

Now I'm sending the project code please take look and please appreciate my work.

For this assignment I need full of two days do develop that mini project and after that half day for testing and documentation.

I'm this assignment I google for some of errors and solutions also w3-school is helped me for this assignment.

But in that case, I write that codes by myself before to take help from resources.

Please also take a look of my Stack overflow account:

<https://stackoverflow.com/users/11067216/akshay-karande>

THANK YOU