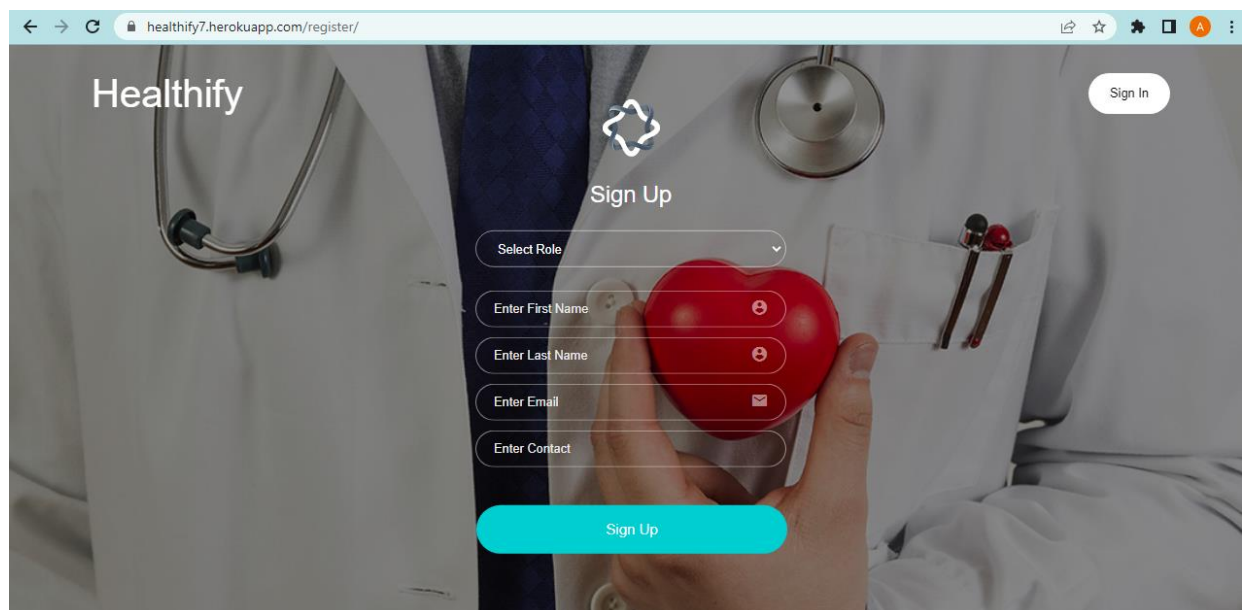


Django Framework Python Assessment Test

- Write a program to demonstrate the Healthify application using of Django web framework
- Prepare demonstration of Healthify 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



□ **Manage below operations**

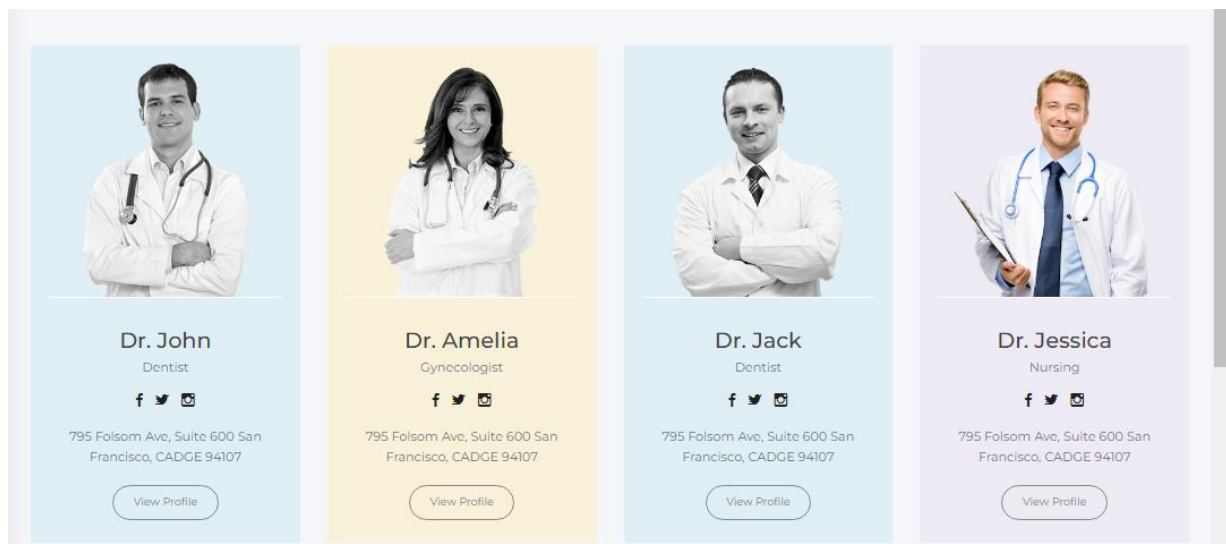
- **Doctor**

- **Registration**
- **Login**
- **Can view other doctors**
- **Can manage (update/delete) all patients appointment request**

- **Patient**

- **Registration**
- **Login**
- **Can View all doctors**
- **Can book appointment to specific doctor**

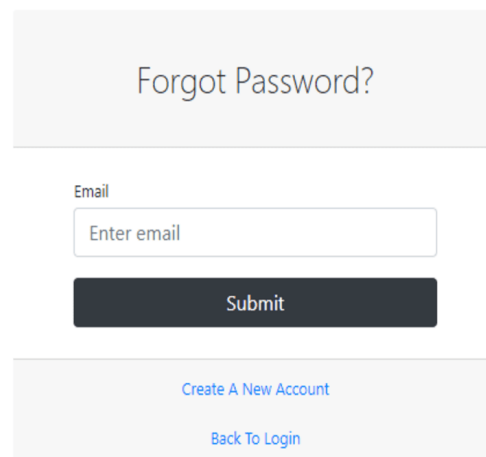
Patient can view all doctor as follows:



Doctor can view all patients list as follows:
Use for loop concept with query set data

Patients List							
All USA India							
Media	Patients ID	Name	Age	Address	Number	Last Visit	Status
	KU 00598	Daniel	32	71 Pilgrim Avenue Chevy Chase, MD 20815	404-447-6013	11 Jan 2018	APPROVED
	KU 00258	Alexander	22	123 6th St. Melbourne, FL 32904	404-447-6013	15 Jan 2018	APPROVED
	KU 00369	Richard	26	123 6th St. Melbourne, FL 32904	404-447-6013	16 Jan 2018	APPROVED
	KU 00147	Samuel	45	71 Pilgrim Avenue Chevy Chase, MD 20815	404-447-6013	17 Jan 2018	APPROVED
	KU 00123	Stephen	55	71 Pilgrim Avenue Chevy Chase, MD 20815	404-447-6013	18 Jan 2018	APPROVED
	KU 00456	Joseph	27	70 Bowman St. South Windsor, CT 06074	404-447-6013	19 Jan 2018	APPROVED
	KU 00789	Cameron	38	123 6th St. Melbourne, FL 32904	404-447-6013	19 Jan 2018	PENDING

- Make sure each business logic is denoted with appropriate comments and make your code interactive and represent clean and clear output on browser.
- Create Django Super admin using Django traditional method – make sure super admin can able to view all user
- 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 Sqlite database.
- Make sure above registration page contain validation using of javascript – store register data using of ajax module.
- After registration send confirmation mail to register email address.
- Email credential and password must store in settings.py and global use in entire the web application
- Apply forgot password feature in this application



A screenshot of a web form titled "Forgot Password?". The form is centered on a light gray background. It features a text input field with the placeholder "Enter email" and a dark gray "Submit" button. Below the form, there are two links: "Create A New Account" and "Back To Login", both in blue text.

- Create separate utils.py for unique business logics
- User can view own profile and able to update it
- Admin can able to delete existing records of this system

- **Manage proper naming convention – 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**

Manage all sessions (Login and Logout)

- **After completion this project upload it on GitHub**
 - **Upload all features in develop branch after completion all features merge it with main branch**

