



LOVELY
PROFESSIONAL
UNIVERSITY

Transforming Education Transforming India

Online Python Interpreter

Language - Python
Framework - Django

Submitted To – Mr. Sagar Pande

Group Members:

- 1) Bhaskar Mahto – 11917448
- 2) Abunachar Yeahhia Mazumder - 11917210

Subject – Python Programming

Code - INT-213

Acknowledgement

We would like to express our special thanks of gratitude to our faculty **Mr. Sagar Pande** who gave us the golden opportunity to do this wonderful and insightful project on the topic **Online python Compiler project**, which also helped us to improve on python language as well as having the practical of the knowledge whatever we learnt in this language. We have gained great experience working in this project and We also got the clear idea of the fact that in how many ways we can use this particular language. We got to know so many new facts and we are really thankful to him.

At last but not the least I want to thanks my friends, classmates and colleagues who help me in completing this Project and for encouraging us for do this types of useful project to boost my knowledge. I have no valuable words to express my thanks but my heart is full of all the favors received from every person.

Summary

There are many resource out there but lack of practice makes thing harder to keep in mind for long term or people don't know where to get started and always lose track form what they are aiming for.

Analyzing this we have created this Project which is a Online Python Compiler website Using Django Framework and Python which provide everyone a open source knowledge resource to learn new Topic of python and practice the topic what they have learn in the same platform.

This a Simple Minimalist site provides a Friendly user interface for the user to interact. There are may many modules Which user can access, like Learn Module in which user can get beginner and advance type question. User can select the Beginner or Advance as per his/her level and see different types questions. To access the answers of the Question User have to login or sign up if they don't have an account. After Successful login the user will be Automatically redirected to the answer page. The login/sign up system is very save with Django password hashing and salting in password, which cannot be decoded without exact key, Salting code and hashing algorithms used. The data are stored in SQLite3 Database.

If user Want to run the program he/she can copy the code directly go to the Compiler module to run the code. The Compiler module has a simple user interface with all placeholder for new user to navigate. There are 3 main panel which are code panel, input panel, terminal/output panel. The code panel in which user can type the code which they want to run ,next is the input panel it the program contain any input syntax they can use the input panel to give all the required input else it can be left empty and the last output panel which is an read only panel which prints all the output and the total time of execution and total memory consumed by the program. These module are achieved using Hacker Rank API module which we have implemented in out project.

The main contributor of this project are **Abunachar Yeahhia** ([knight-byte on GitHub](#)) and **Bhaskar Mahto** ([arbk22 on GitHub](#))

Index

- ❖ Introduction
- ❖ Review of Literature
- ❖ Project Contribution
 - Material and Methods
 - Result
 - Conclusion
- ❖ Bibliography

Introduction

This Project which is a Online Python Compiler website Using Django Framework and Python which provide everyone a open source knowledge resource to learn new Topic of python and practice the topic what they have learn in the same platform.

This a Simple Minimalist site provides a Friendly user interface for the user to interact. There are may many modules Which user can access, like Learn Module in which user can get beginner and advance type question. User can select the Beginner or Advance as per his/her level and see different types questions. To access the answers of the Question User have to login or sign up if they don't have an account. After Successful login the user will be Automatically redirected to the answer page. The login/sign up system is very save with Django password hashing and salting in password, which cannot be decoded without exact key, Salting code and hashing algorithms used. The data are stored in SQLite3 Database.

If user Want to run the program he/she can copy the code directly go to the Compiler module to run the code. The Compiler module has a simple user interface with all placeholder for new user to navigate. There are 3 main panel which are code panel, input panel, terminal/output panel. The code panel in which user can type the code which they want to run ,next is the input panel it the program contain any input syntax they can use the input panel to give all the required input else it can be left empty and the last output panel which is an read only panel which prints all the output and the total time of execution and total memory consumed by the program. These module are achieved using Hacker Rank API module which we have implemented in out project.

Review of Literature

Python is an interpreted, high-level and general-purpose programming language. Created by Guido van Rossum and first released in 1991, Python's design philosophy emphasizes code readability with its notable use of significant whitespace. Its language constructs and object-oriented approach aim to help programmers write clear, logical code for small and large-scale projects.

Python is dynamically typed and garbage-collected. It supports multiple programming paradigms, including structured (particularly, procedural), object-oriented, and functional programming. Python is often described as a "batteries included" language due to its comprehensive standard library.

Python was created in the late 1980s as a successor to the ABC language. Python 2.0, released in 2000, introduced features like list comprehensions and a garbage collection system with reference counting.

Python 3.0, released in 2008, was a major revision of the language that is not completely backward-compatible, and much Python 2 code does not run unmodified on Python 3.

Django is a Python-based free and open-source web framework that follows the model-template-views (MTV) architectural pattern. It is maintained by the Django Software Foundation (DSF), an American independent organization established as a 501(c)(3) non-profit.

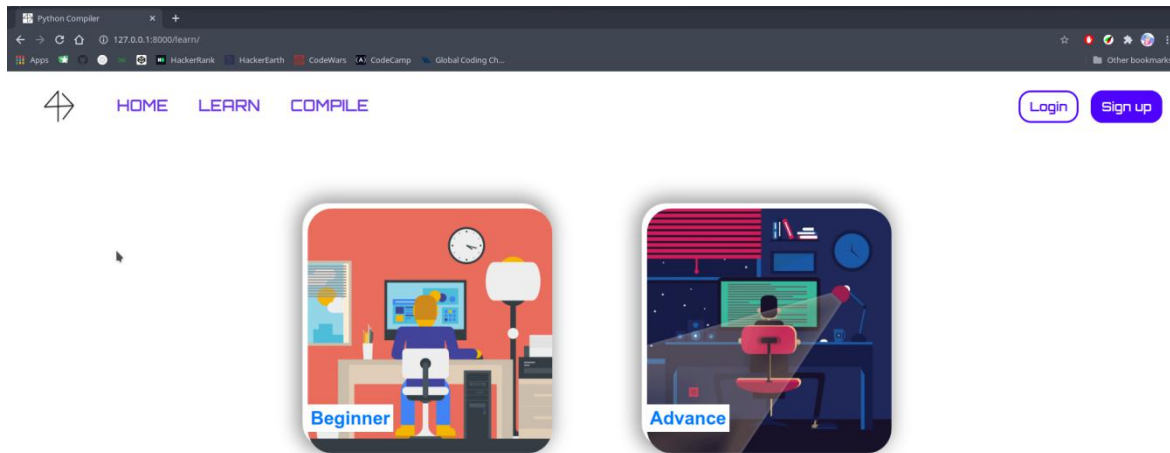
Django's primary goal is to ease the creation of complex, database-driven websites. The framework emphasizes reusability and "pluggability" of components, less code, low coupling, rapid development, and the principle of don't repeat yourself. Python is used throughout, even for settings files and data models. Django also provides an optional administrative create, read, update and delete interface that is generated dynamically through introspection and configured via admin models.

HOME PAGE



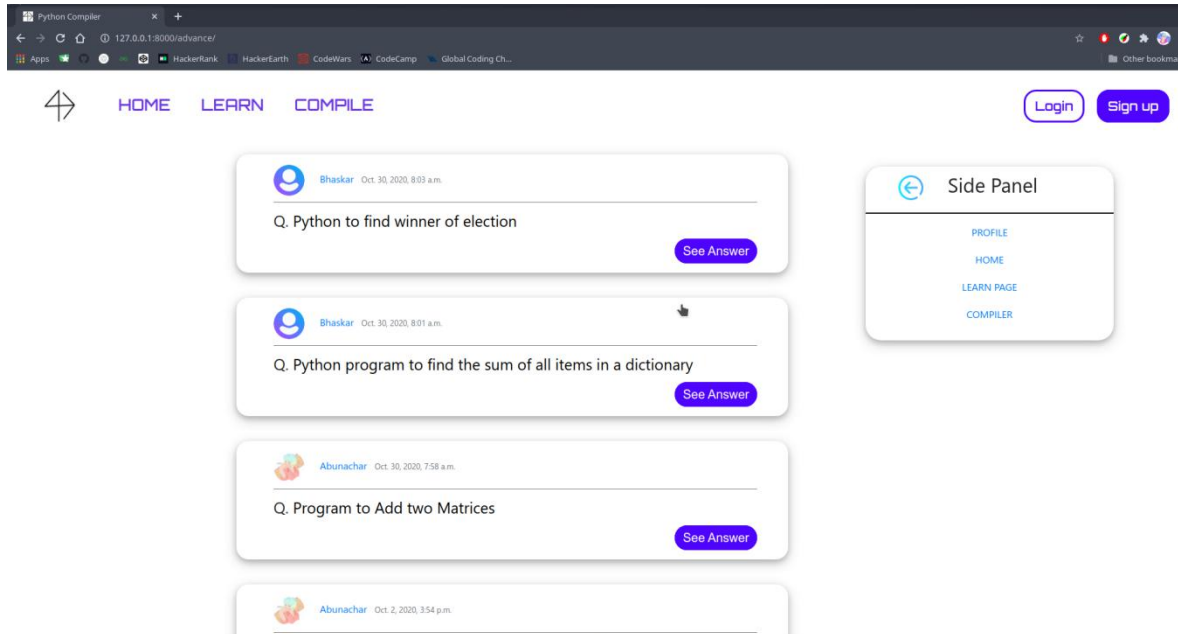
Home page contain a Quote by Guido Van Rossum about python programming and side sliding animation of slides. Top Navigation bar contain all the required hyperlink to navigate all over the site.

LEARN PAGE



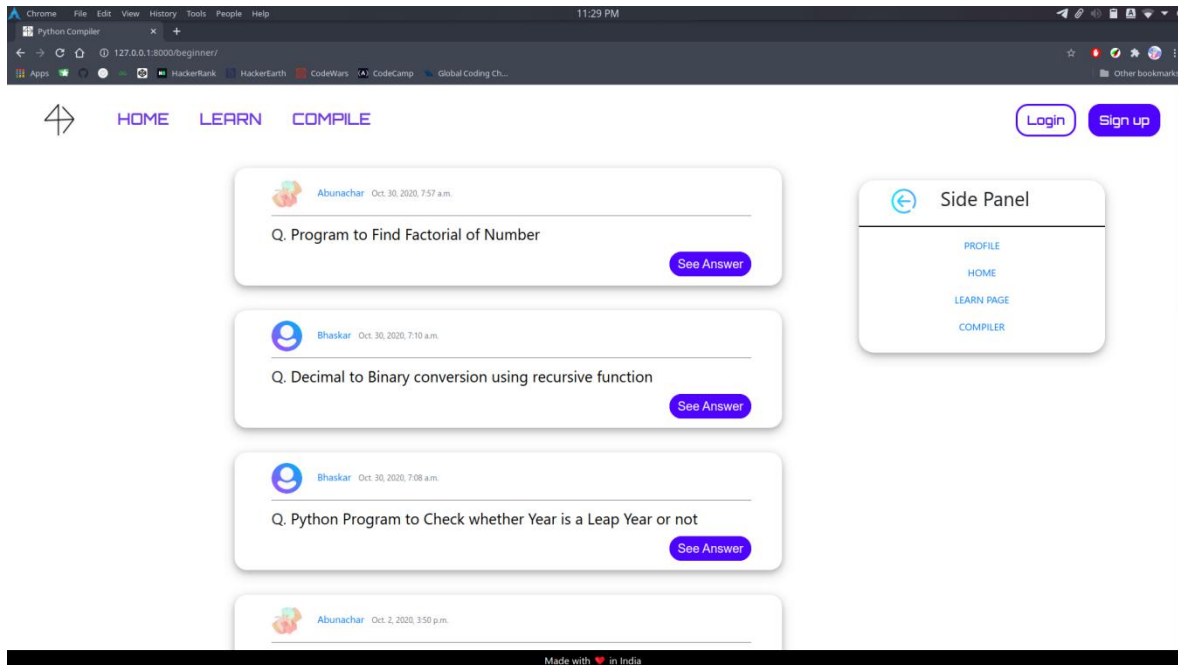
Learn page contain two category of post First one is beginner post and second on is the Advance post User can choose from here what type of programs/Question they want to solve.

BEGINNER POST PAGE



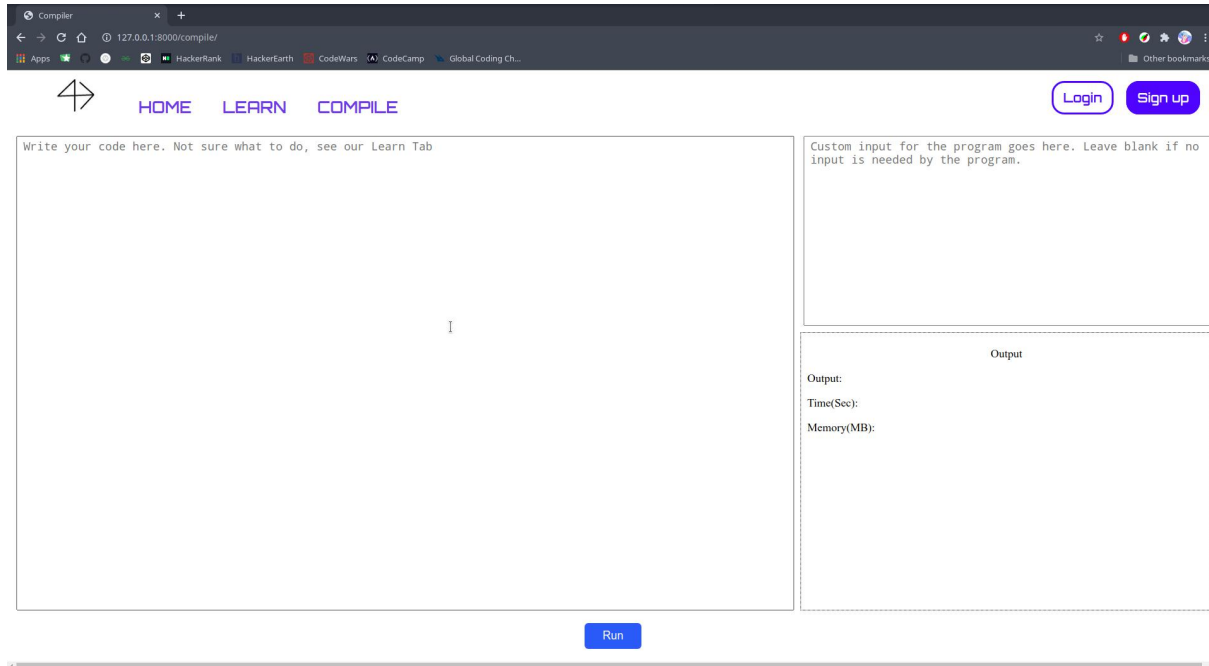
After selecting the Beginner Post option in Learn page User can see different types of Question. if they want to view the Solution they have to login first to view. User can navigate with the side panel provided to the side for creating/update/delete post (if they are staff or authorized person)

ADVANCE POST PAGE



After selecting the Advance Post option in Learn page User can see different types of Question. if they want to view the Solution they have to login first to view. User can navigate with the side panel provided to the side for creating/update/delete post (if they are staff or authorized person)

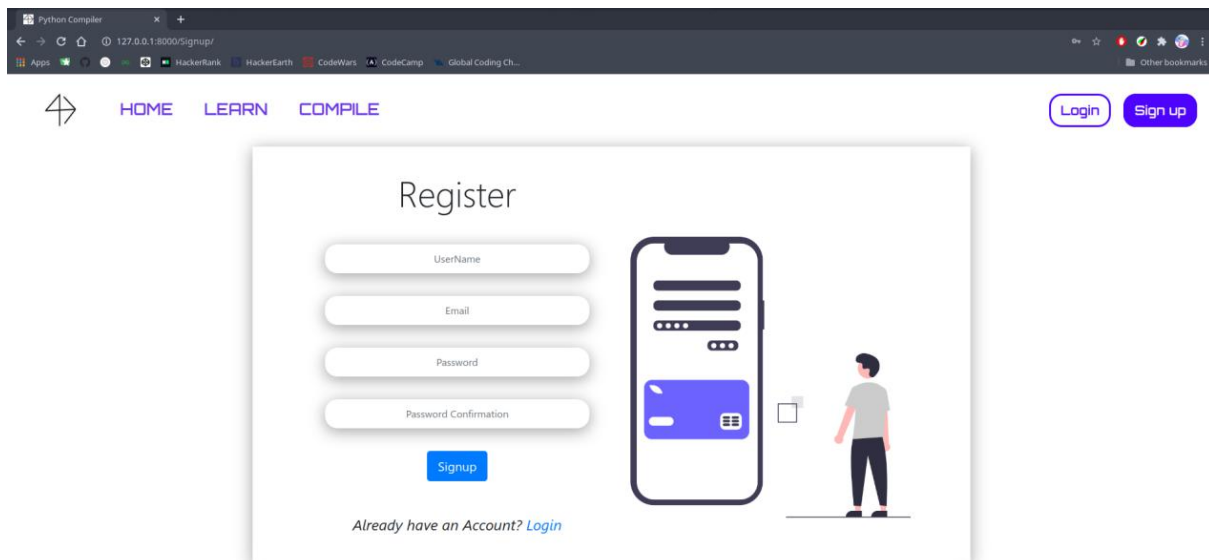
COMPILER PAGE



The screenshot shows a web browser window with the URL `127.0.0.1:8000/compiler/`. The page has a navigation bar with links for [HOME](#), [LEARN](#), and [COMPILE](#), along with [Login](#) and [Sign up](#) buttons. The main content area is divided into three panels: a large code editor on the left with the placeholder text "Write your code here. Not sure what to do, see our Learn Tab"; a custom input panel on the top right with the text "Custom input for the program goes here. Leave blank if no input is needed by the program."; and an output panel on the bottom right labeled "Output" with fields for "Output:", "Time(See):", and "Memory(MB):". A blue "Run" button is positioned at the bottom center of the interface.

User can run there program here. This compiler works on HackerRank API v3. user can see three Panel code panel for all coding/program user want to run. second one is input panel if user want to insert any input in the program else it can be kept empty and the last one is the Terminal panel to view the output/result of the program with total time taken and memory consumed by the program.

SIGIN UP PAGE

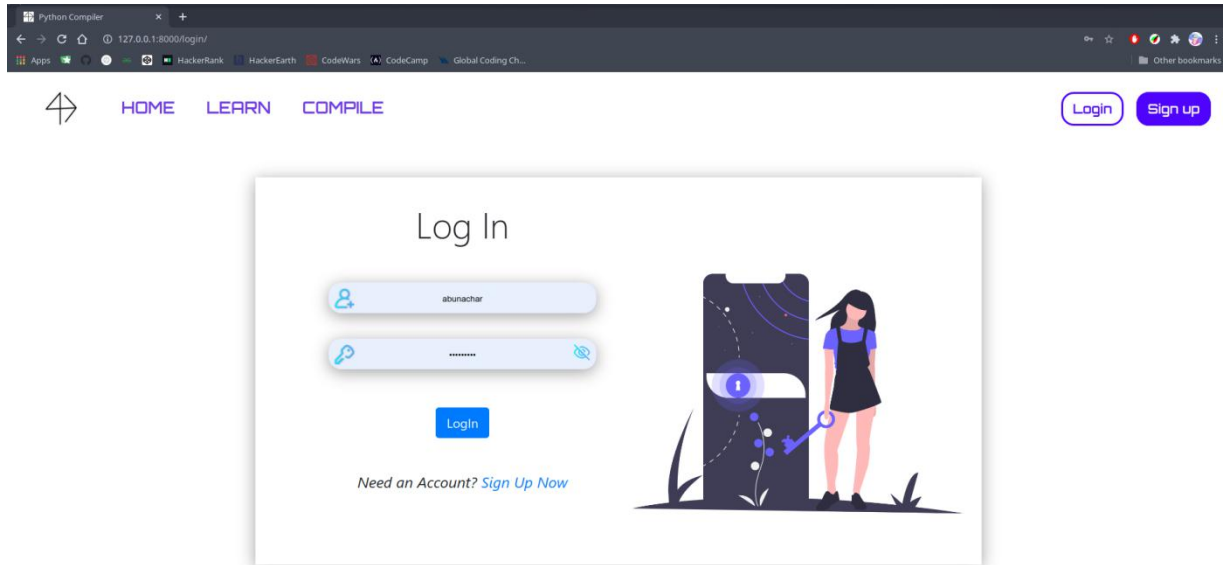


The screenshot shows a web browser window with the URL `127.0.0.1:8000/signup/`. The page features a navigation bar with [HOME](#), [LEARN](#), and [COMPILE](#) links, and [Login](#) and [Sign up](#) buttons. The main content area displays a "Register" form with input fields for "UserName", "Email", "Password", and "Password Confirmation". A blue "Signup" button is located below the form. To the right of the form is an illustration of a smartphone and a person. At the bottom of the form, there is a link: "Already have an Account? [Login](#)".

Made with  in India

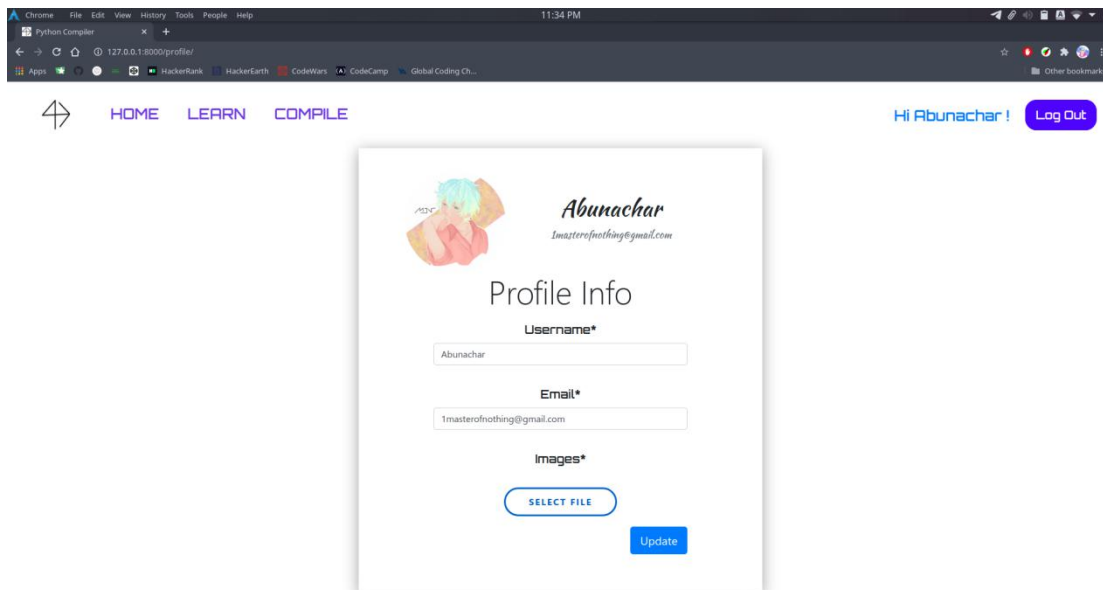
User Can signup in the Signup page, The user name will be Unique for every User. User detail are stored in the server and password are encrypted by django hashing method with salting. user will be redirected to login after successful register.

LOGIN PAGE



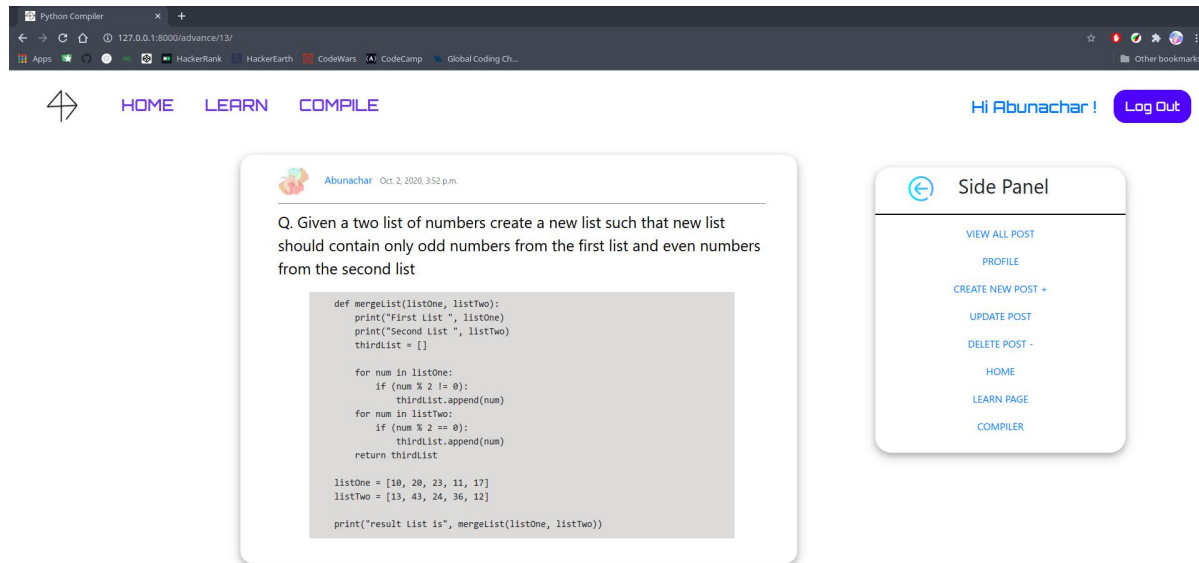
This is the login page User Registered user can login with there Login Username and Password.After successful login user are redirected to the Home page.

USER PROFILE PAGE



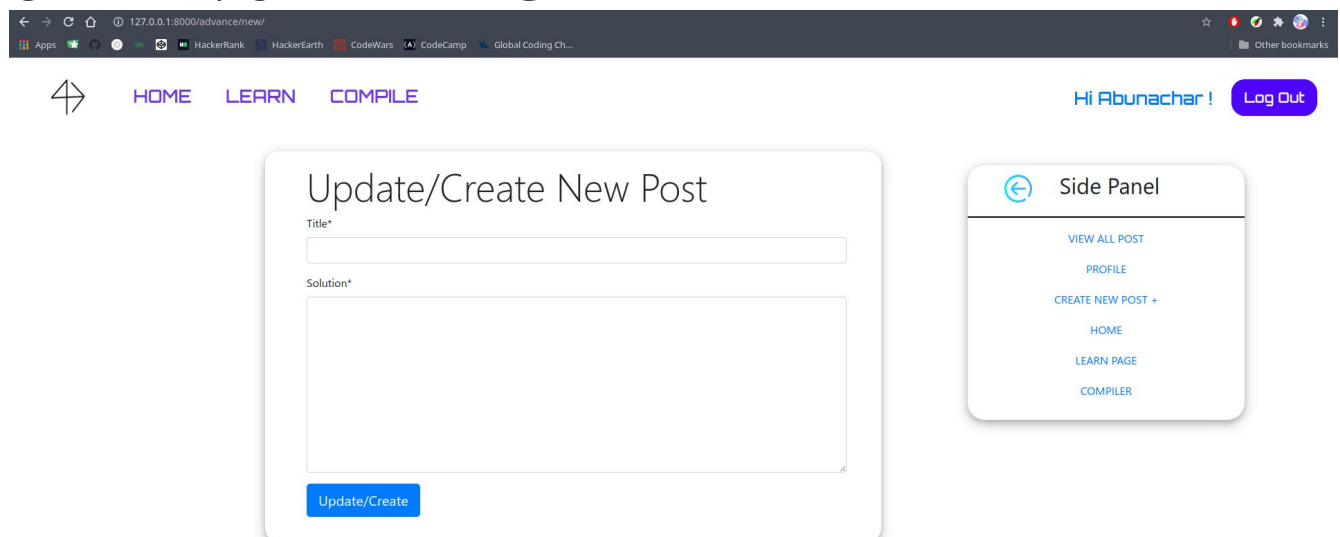
This is the User Profile where user can Add profile image and update their profile information.

DETAILED POST/ANSWER PAGE



This is the Detailed post/Answer page where user can view the Full Question and answer. They can only access this page if they have successfully login in with an account

CREATE / UPDATE PAGE



This page is accessible Only to the Staff member of the page where they can Update existing post or create a new post.

MATERIALS & METHODS

The following list showcases all the materials used for making the project i.e., Online Python Compiler.

- Django (v2.1): The main Python Framework, used to handle the Front-End as well as the Back-End of the project.
- requests(v2.24.0): Python library for handling all the HTTP requests made from the user side.
- djangorestframework(v3.12.1): Python module for django, to handle all the rest-api calls and POST/GET requests.
- django-crispy-forms(v1.9.2): Django module for better form customisation.
- HackerEarth API(v3): API for compiling and running code on HackerEarth servers instead of using user's system.

RESULT

The final outcome of our project is a fully functional python interpreter that is not system dependent. It runs on the cloud and doesn't require your system to have python installed. It can also work on handheld devices like Smartphones or even PSP's (as long as the device have a compatible browser that supports ES6). The only drawback is the lack of dynamic programming, i.e., the user can't interact with the system once the program has been submitted. If there are any inputs that is required for the program to run, then the user has to submit it beforehand.

Also, we have a integrated learning section, that provides with some good examples based on the level of experience of the user, i.e., beginner to advancewd. We have provided some common examples and their solution too.

We have also integrated a user database for more functionalities like the option to add your own set of examples so that other users can also benefit from it. To add or view answers, you need to be registered on the site.

To protect the server from heinous inputs, we have added CSRF_Tokens, so that the security of the site remains uncompromised, along with the user data.

DISCUSSION (including conclusion, recommendation)

The final outcome of the program is a clean and proper working Python Interpreter. We have hosted the web app on Heroku. Since we believe that Open Source is the way of the future, the source code of the app is made publicly available on GitHub along with the API Token, so that our users, if they find any errors, or want to contribute to it, can do so freely without having to worry about the API Tokens (as it is very cumbersome to get the Tokens). If you want to make any improvements, or have any suggestions, fork the repo, and create a pull request. We will review and merge the request ASAP.

To run and test the app locally, follow the instructions on our GitHub readme.

Recommendation

Some users may want to use Flask instead of Django, as Flask is more lightweight and gives the user more power and authority over the app. To be honest, we intended to use Flask for this, but due to some difficulties, we had to settle in with Django. The main api handler can be found in the “views.py” inside “compiler” directory of the main program.

Bibliography

Web Links :

- <https://docs.python.org/3/library/index.html>
- <https://docs.djangoproject.com/en/3.1>
- <https://www.hackerrank.com/api/docs>

Book and Reference :

- Django for Beginners: Build website with python and Django by William S. Vincent