

**Misr University for Science & Technology**

**College of Information Technology**

|  |  |
| --- | --- |
| 94289 | Kyrillos Emad |

**Title: Blog.**

**Supervised by: Dr. Khaled Elsheshtawy**

**TA: Islam Saeed**

**Fall 2021 / 2022**

**The index**

|  |  |
| --- | --- |
| **Introduction** | Page 3 |
| **Using python** | Page 4 |
| **Using HTML, CSS, JavaScript** | Page 5 |
| **The running program(GUI)** | Page 7 |
| **The code** | Page 14 |
| **The diagrams** | Page 18 |
|  | |

**Introduction**

X-BLOG is a website that provides free creating posts; my project blog is a weblog - a sort of Internet diary - that charts the progress, status, and details of a certain defined endeavor. These blogs can come in many forms. Some are designed as a way for corporate team members to update each other on progress made on a certain work project. More common is a public project blog, which acts as an outward interfacing tool between producers, clients, and the interested public. A project blog can also be an individual account of a personal project or challenge.  
  
A blog post is an individual web page on your website that dives into a particular sub-topic of your blog.

For instance, let's say you start a fashion blog on your retail website. One blog post might be titled, "The Best Fall Shoes for 2019". The post ties back to your overall blog topic as a whole (fashion), but it also addresses a very particular sub-topic (fall shoes).

Blog posts allow you to rank on search engines for a variety of keywords. In the above example, your blog post could enable your business to rank on Google for "fall shoes". When someone searches for fall shoes and comes across your blog post, they have access to the rest of your company's website. They might click "Products" after they read your post, and take a look at the clothing items your company sells.

Running a blog isn’t a piece of cake. Successful bloggers may make it look easy, but we rarely see what’s going on behind the scenes.

Behind the great graphics and snappy writing, is a whole heap of stress, hustle and long working days.

For a blog to thrive, expect yourself to be working more hours than a standard day job. And those hours often extend into the evenings and weekends when you’d normally be relaxing.

**Why we used python?**

Python is a widely used general-purpose, high level programming language. It was **created by Guido van Rossum in 1991** and further developed by the Python Software Foundation. It was designed with an emphasis on code readability, and its syntax allows programmers to express their concepts in fewer lines of code.  
**Why we used Django?**

Django is a high-level Python web framework that enables rapid development of secure and maintainable websites. Built by experienced developers, Django takes care of much of the hassle of web development, so you can focus on writing your app without needing to reinvent the wheel.

**Why Django with Python?**

Django is an extremely popular and fully featured server-side web framework, written in Python. This module shows you why Django is one of the most popular web server frameworks, how to set up a development environment, and how to start using it to create your own web applications.

**Why HTML?**

HTML is short for Hypertext Markup Language. HTML is used to create electronic documents (called pages) that are displayed on the World Wide Web. Each page contains a series of connections to other pages called hyperlinks. Every web page you see was written using one version of HTML.

HTML code ensures the proper formatting of text and images for your Internet browser. Without HTML, a browser would not know how to display text as elements or load images or other elements. HTML also provides a basic structure of the page, upon which Cascading Style Sheets are overlaid to change its appearance. One could think of HTML as the bones (structure) of a web page, and CSS as its skin (appearance).

**Why CSS ?**

CSS helps you to keep the informational content of a document separate from the details of how to display it. The details of how to display the document are known as its style. You keep the style separate from the content so that you can:

Avoid duplication

Make maintenance easier

Use the same content with different styles for different purposes

Your web site might have thousands of pages that look similar. Using CSS, you store the style information in common files that all the pages share. When a user displays a web page, the user’s browser loads the style information along with the content of the page. When a user prints a web page, you might provide different style information that makes the printed page easy to read.

In general, you use HTML to describe the content of the document, not its style; you use CSS to specify its style, not its content. There are exceptions to this rule, of course, and HTML also provides some ways to specify style. For example, in HTML you can use a <b> tag to make text bold, and you can specify the background color of a page in its <body> tag. When you use CSS, you normally avoid using these HTML style features so that all your document’s style information is in one place.

**Why JavaScript?**

JavaScript is a scripting or programming language that allows you to implement complex features on web pages — every time a web page does more than just sit there and display static information for you to look at — displaying timely content updates, interactive maps, animated 2D/3D graphics, scrolling video jukeboxes, etc. — you can bet that JavaScript is probably involved. It is the third layer of the layer cake of standard web technologies, two of which (HTML and CSS) we have covered in much more detail in other parts of the Learning Area.

**Our Project Consist of two apps:**

1. Posts(main page)
2. Account

The Home page consists of:

* Home
* Create post
* My posts
* Profile
* Register
* Login

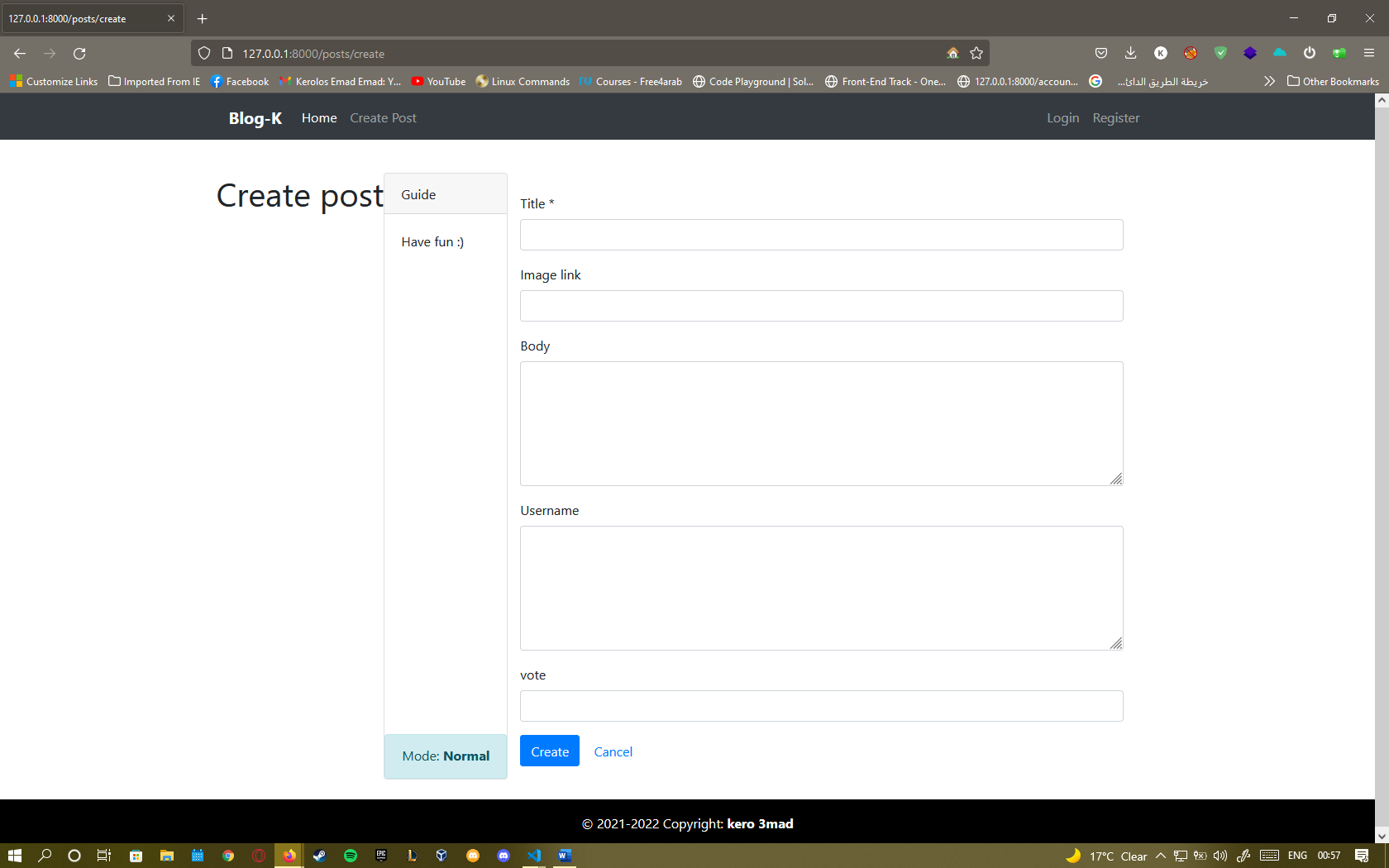
**The running program(GUI)**

The home page:

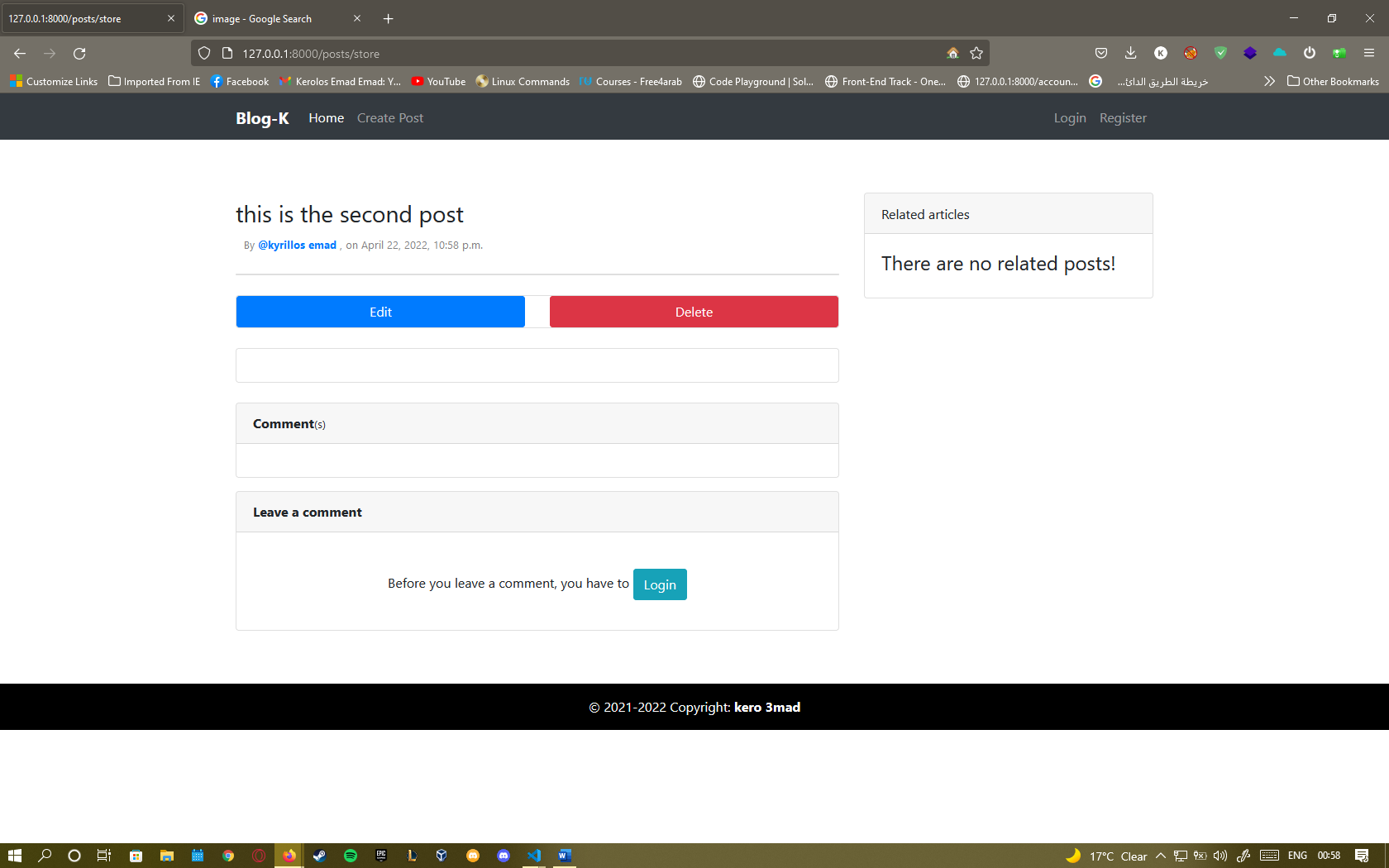
A screenshot of a computer

Description automatically generated

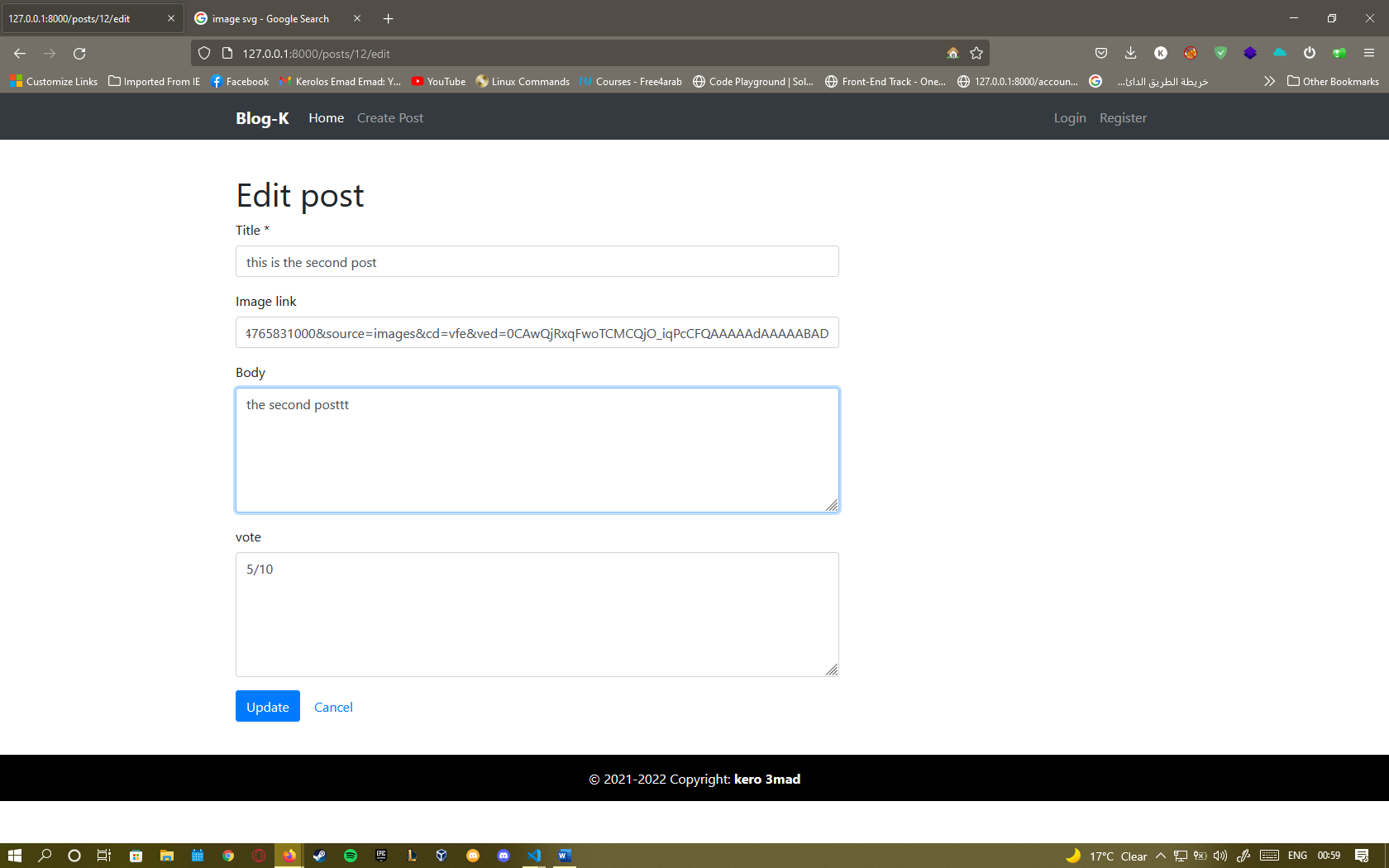
Creating post page:



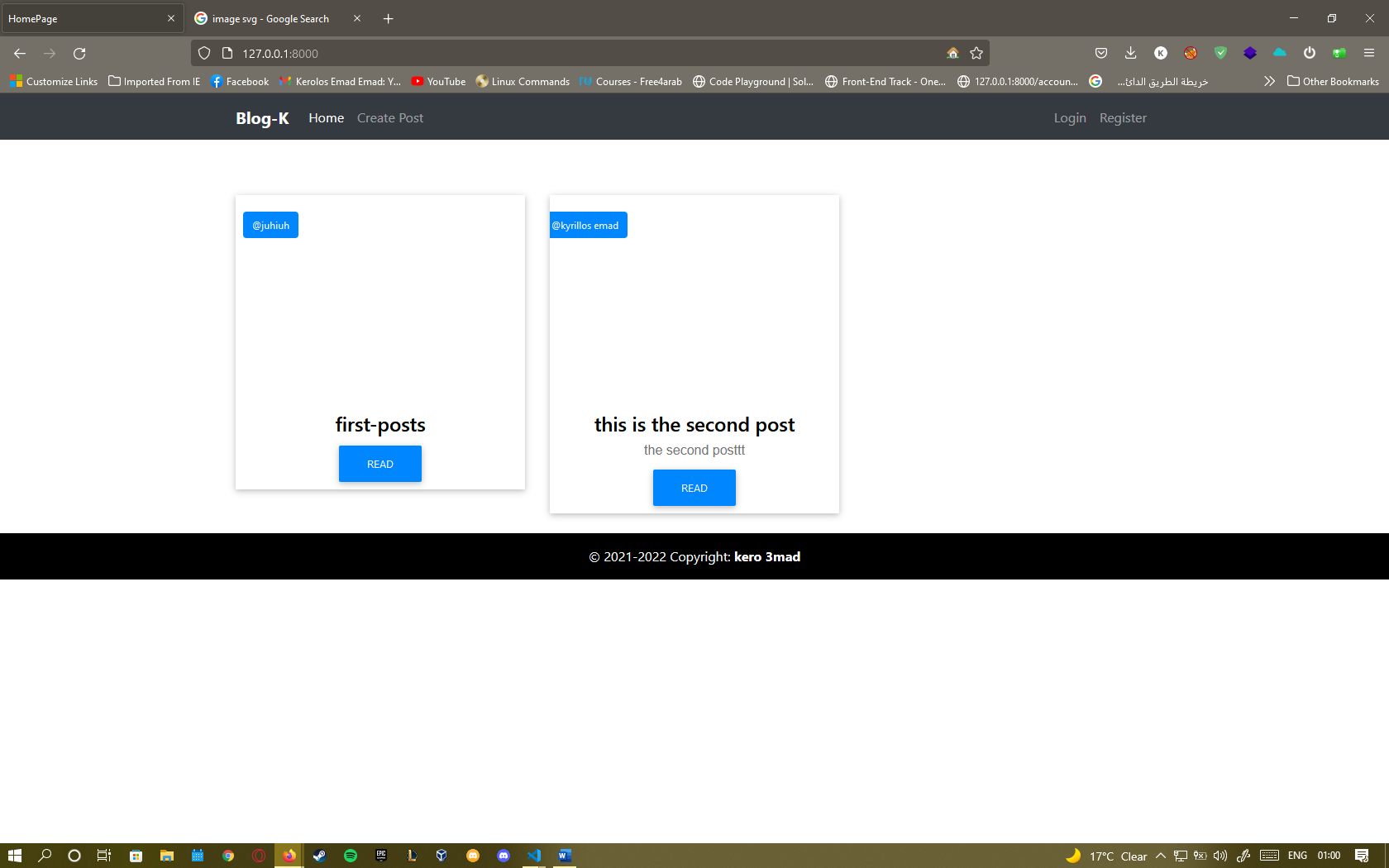
Read page:



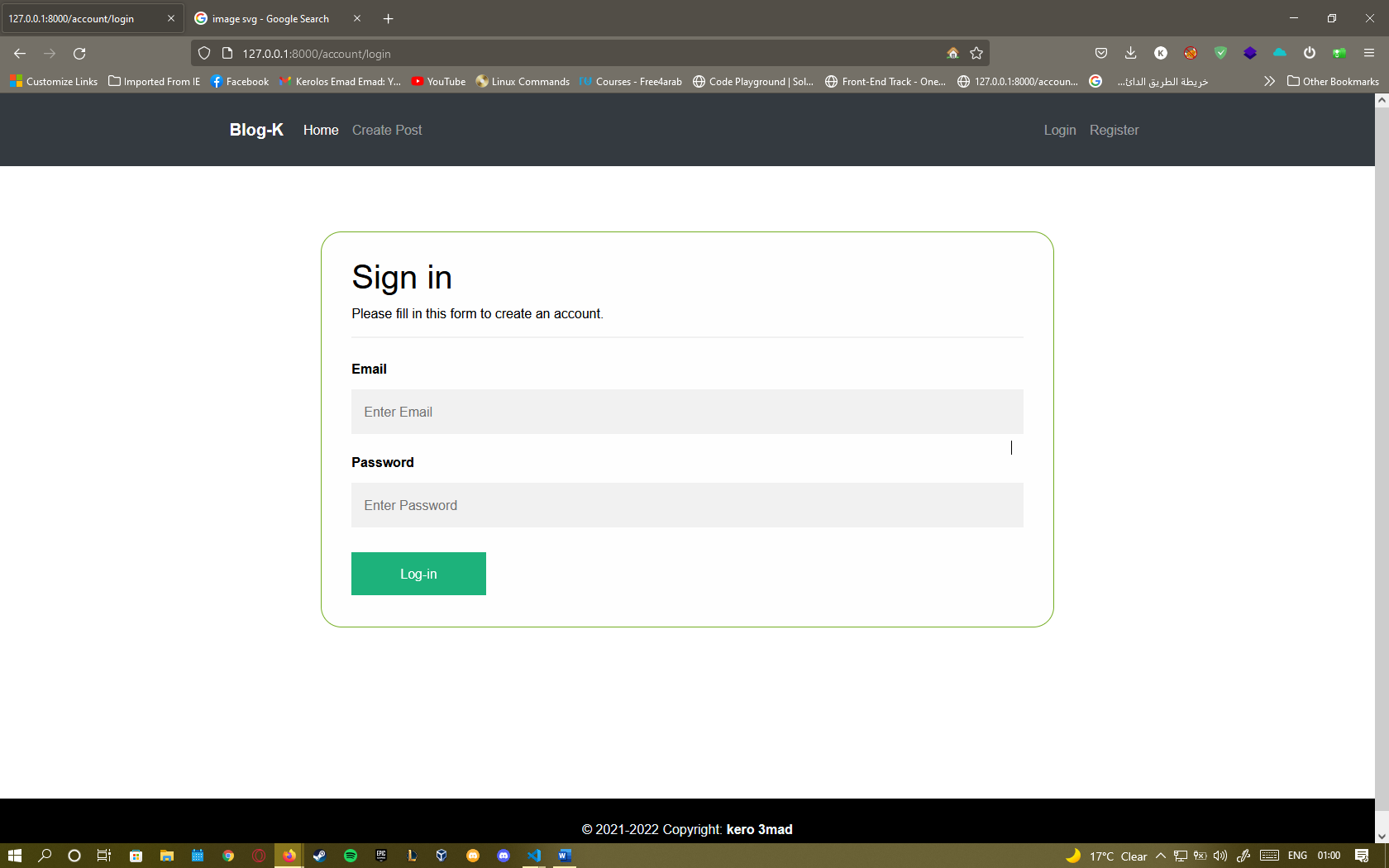
Editing posts page:



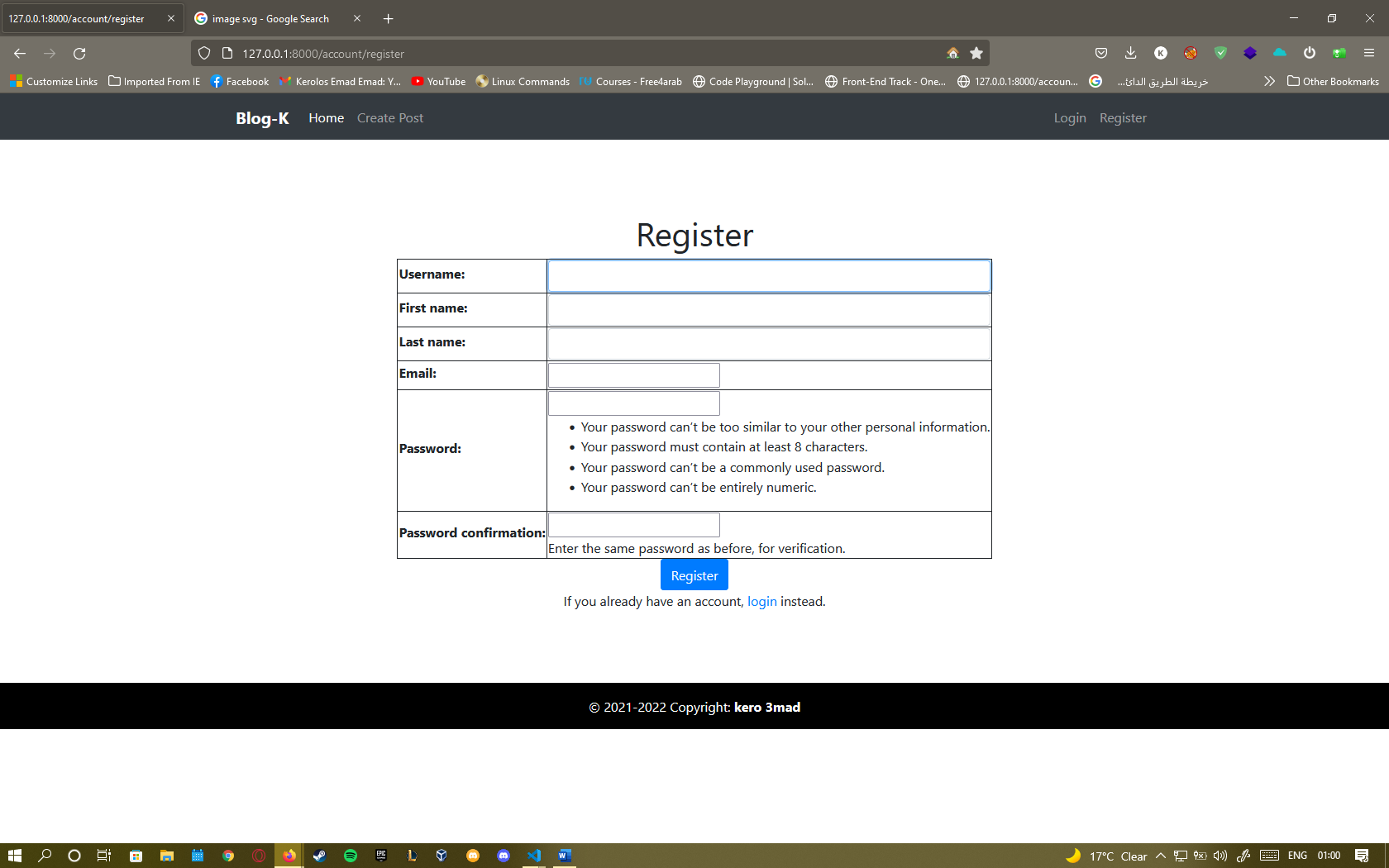
The home page after editing the post:



Login page:

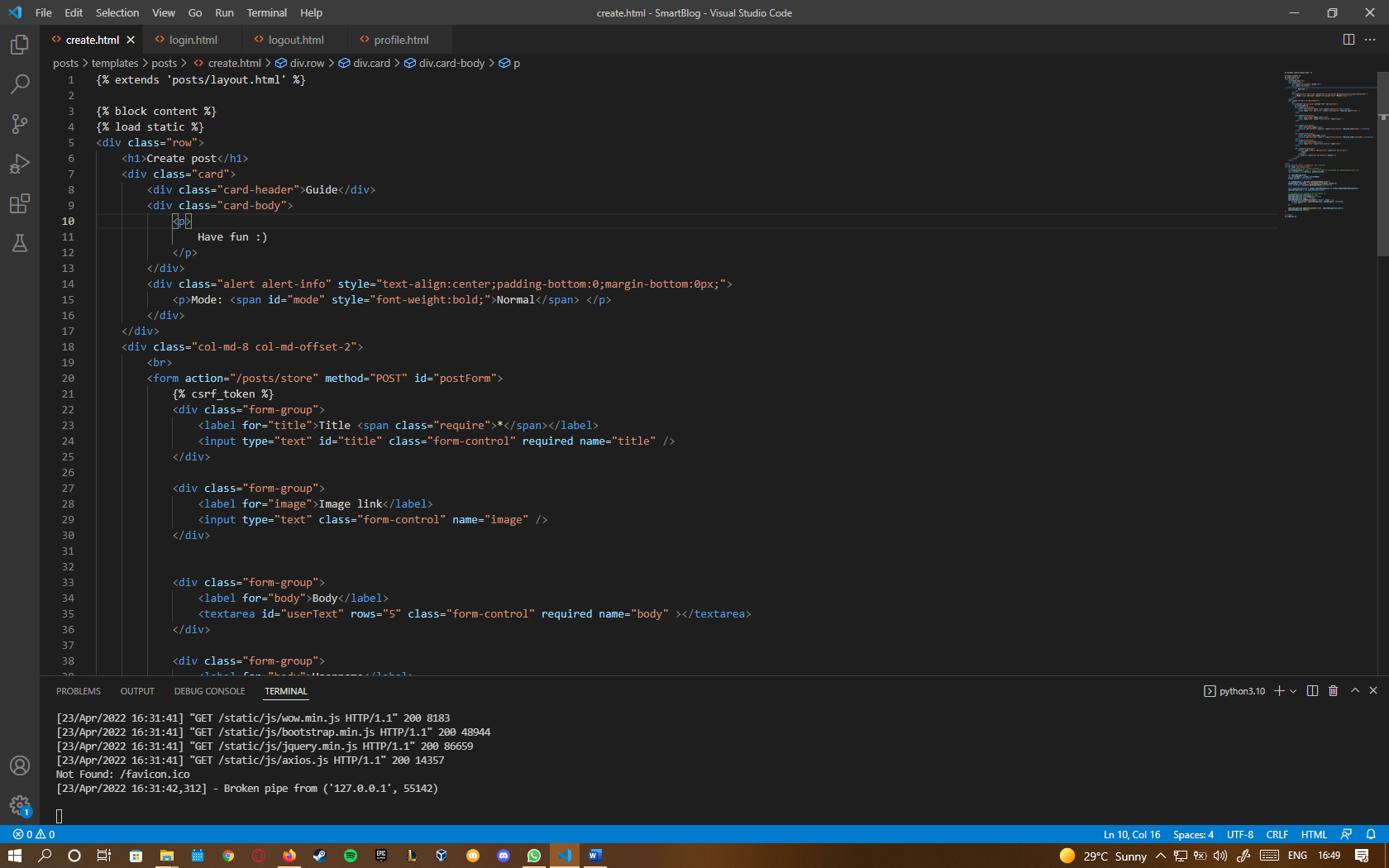


Register page:

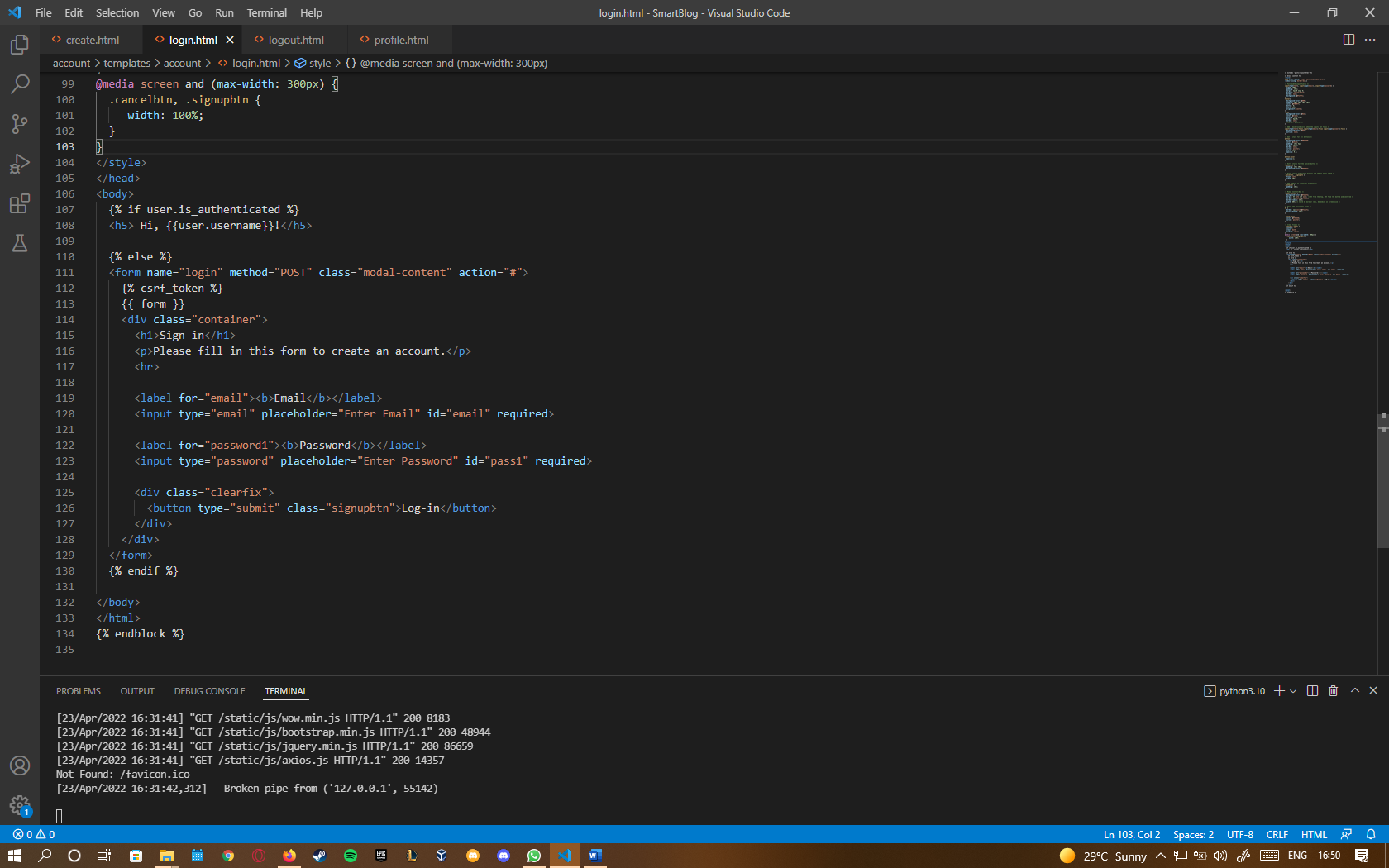


**The code**

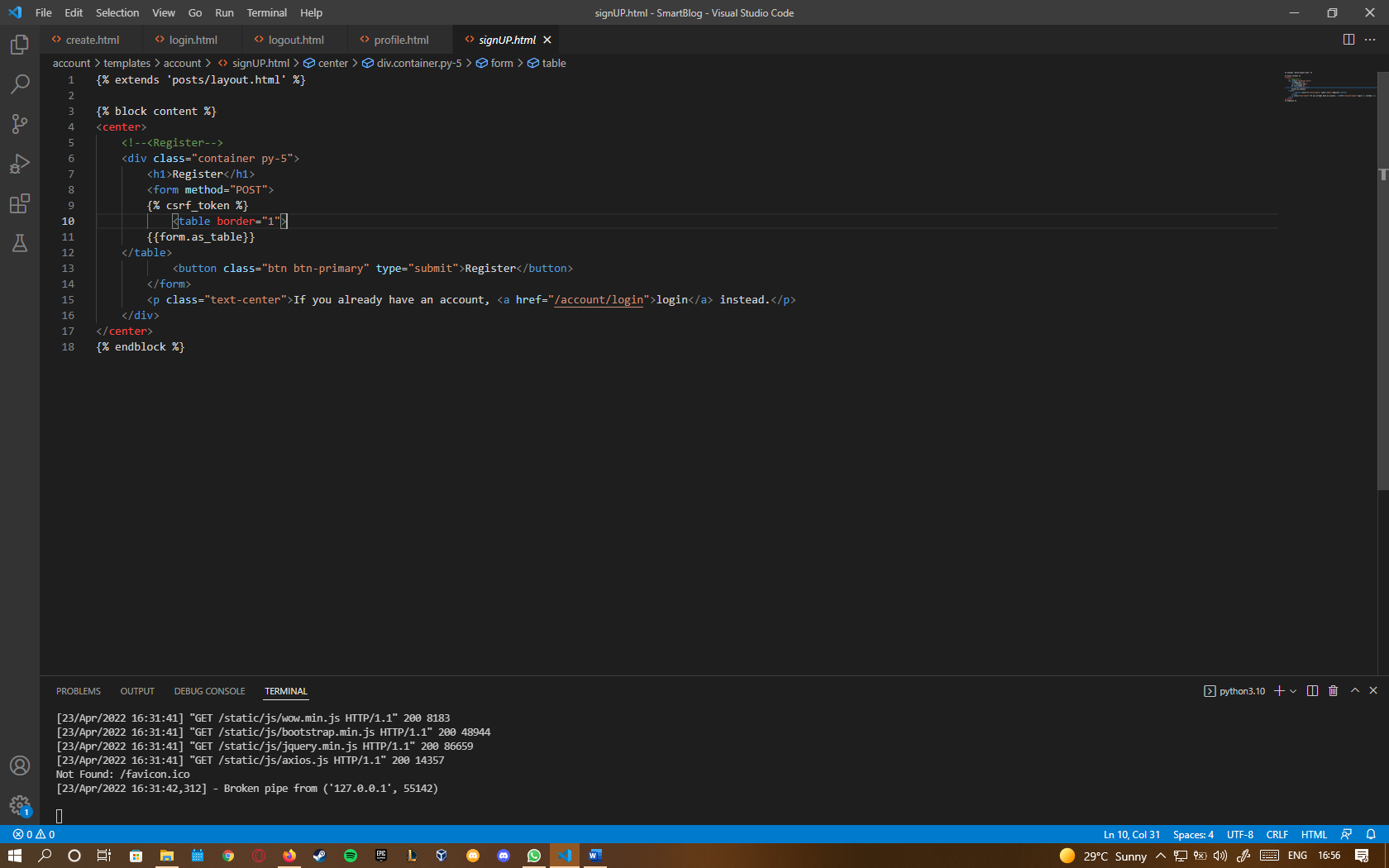
The create page:



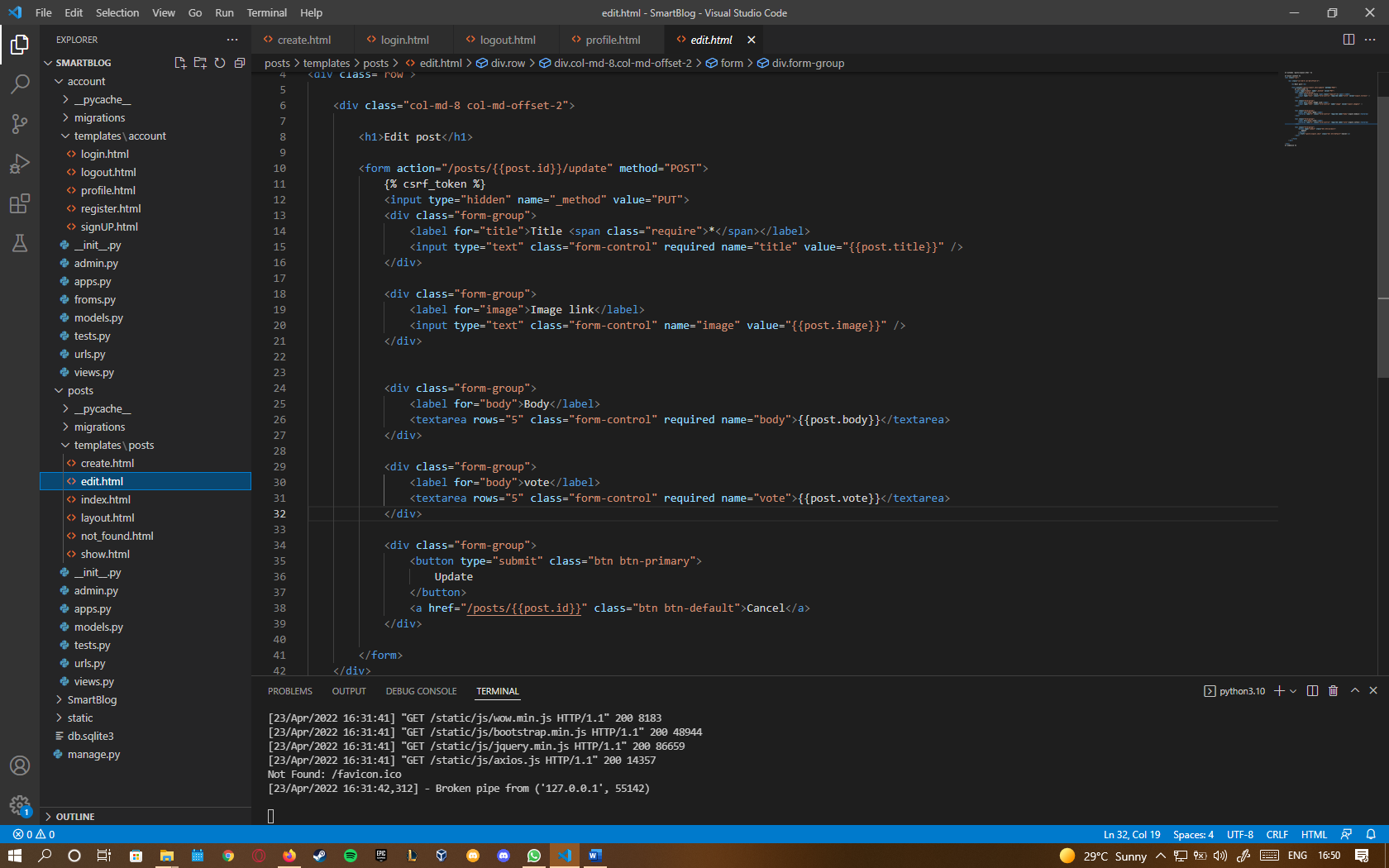
The login page:



The edit page:



The register page:



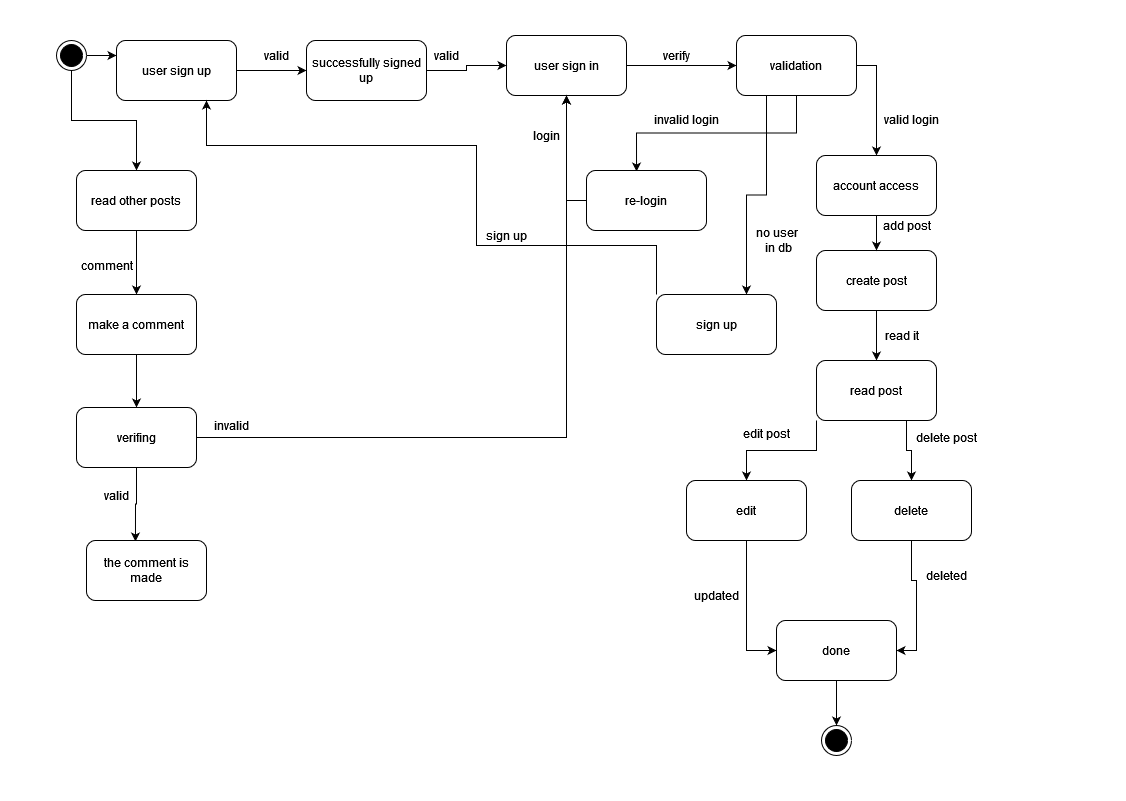
**The diagrams**

Activity diagram:

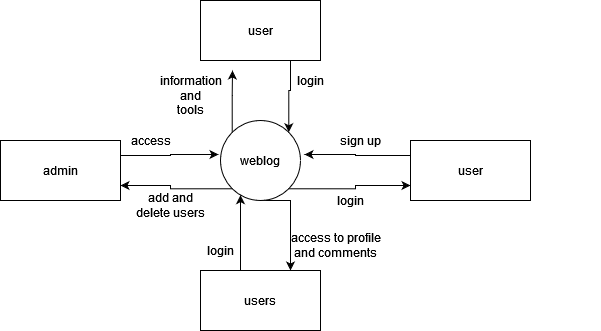
Diagram

Description automatically generated

State diagram:



Context diagram:



Class diagram (with relationships):

