Web Application with Database Integration and Apache Hosting

Prepared By:
Mahmoud Alsalloum

Table of Contents

Contents

Technology Stack

Apache Web Server – Hosts the HTML / CSS based website and executes Python CGI scripts.

MongoDB – NoSQL database used to store customer form submissions. Chosen because the project doesn’t require data. Each customer entry is standalone.

Python3 (CGI) – Handles the server-side logic to receive form input and insert it into MongoDB.

HTML / CSS – Builds a single-page, modern-looking form for user input.

Ubuntu Server – The Linux environment hosting the entire stack.

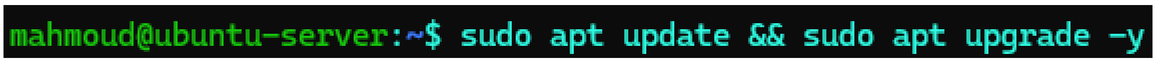
Project Overview

Welcome to my project!

In this small web application, I built a simple system for an IT company where customers can enter their contact details and describe their issue through a web form. Once submitted, their data is saved directly into a MongoDB database on the server. The goal was to keep everything lightweight and easy to understand. No PHP, no JavaScript, no complex frameworks. Just HTML, CSS, Python, and MongoDB, all running an Apache web server. This project shows how to collect, store, and manage user data in a clean and reliable way using basic but powerful tools.

Implementation Steps

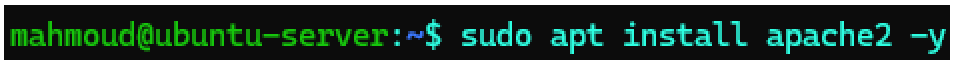
The first step was to update the system packages using the following command in Figure 1.



```
mahmoud@ubuntu-server:~$ sudo apt update && sudo apt upgrade -y
```

Figure 1: Update the system

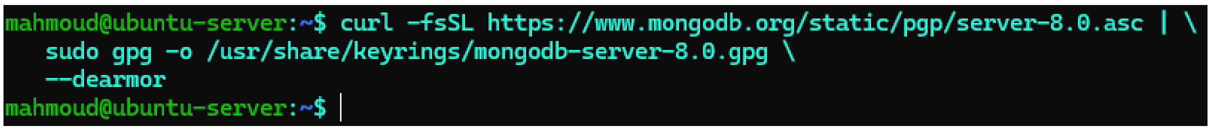
Next, I installed the Apache web server, which will be used to host the web application. See Figure 2.



```
mahmoud@ubuntu-server:~$ sudo apt install apache2 -y
```

Figure 2: Install Apache web server

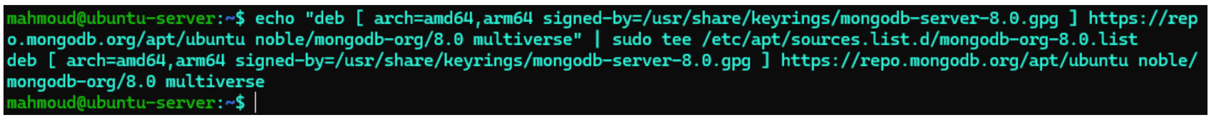
MongoDB was not available in the default Ubuntu repositories, so I had to manually add the official MongoDB repository to install the correct version. I first added the public GPG key used by MongoDB to verify packages. See Figure 3.



```
mahmoud@ubuntu-server:~$ curl -fsSL https://www.mongodb.org/static/pgp/server-8.0.asc | \
   sudo gpg -o /usr/share/keyrings/mongodb-server-8.0.gpg \
   --dearmor
mahmoud@ubuntu-server:~$
```

Figure 3: Adding GPG key

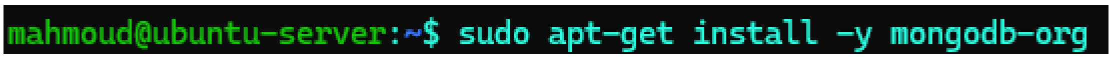
Then I created a new source list file pointing to the official MongoDB repository. See Figure 4.



```
mahmoud@ubuntu-server:~$ echo "deb [ arch=amd64,arm64 signed-by=/usr/share/keyrings/mongodb-server-8.0.gpg ] https://rep
o.mongodb.org/apt/ubuntu noble/mongodb-org/8.0 multiverse" | sudo tee /etc/apt/sources.list.d/mongodb-org-8.0.list
deb [ arch=amd64,arm64 signed-by=/usr/share/keyrings/mongodb-server-8.0.gpg ] https://repo.mongodb.org/apt/ubuntu noble/
mongodb-org/8.0 multiverse
mahmoud@ubuntu-server:~$
```

Figure 4: Adding new Source list file for MongoDB

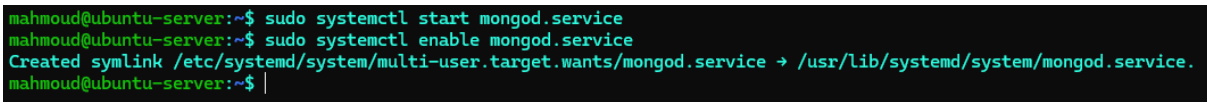
After adding the key and repository, I updated the package list and installed MongoDB. See Figure 5.



```
mahmoud@ubuntu-server:~$ sudo apt-get install -y mongodb-org
```

Figure 5: Installing MongoDB

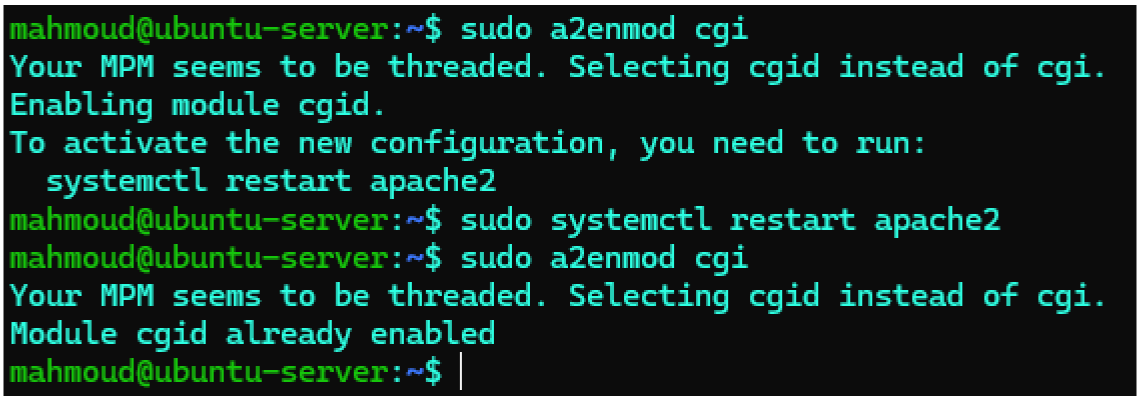
Once installed, I started the MongoDB service and enabled it to run at boot. See Figure 6.



```
mahmoud@ubuntu-server:~$ sudo systemctl start mongod.service
mahmoud@ubuntu-server:~$ sudo systemctl enable mongod.service
Created symlink /etc/systemd/system/multi-user.target.wants/mongod.service → /usr/lib/systemd/system/mongod.service.
mahmoud@ubuntu-server:~$
```

Figure 6: Starting and enabling MongoDB

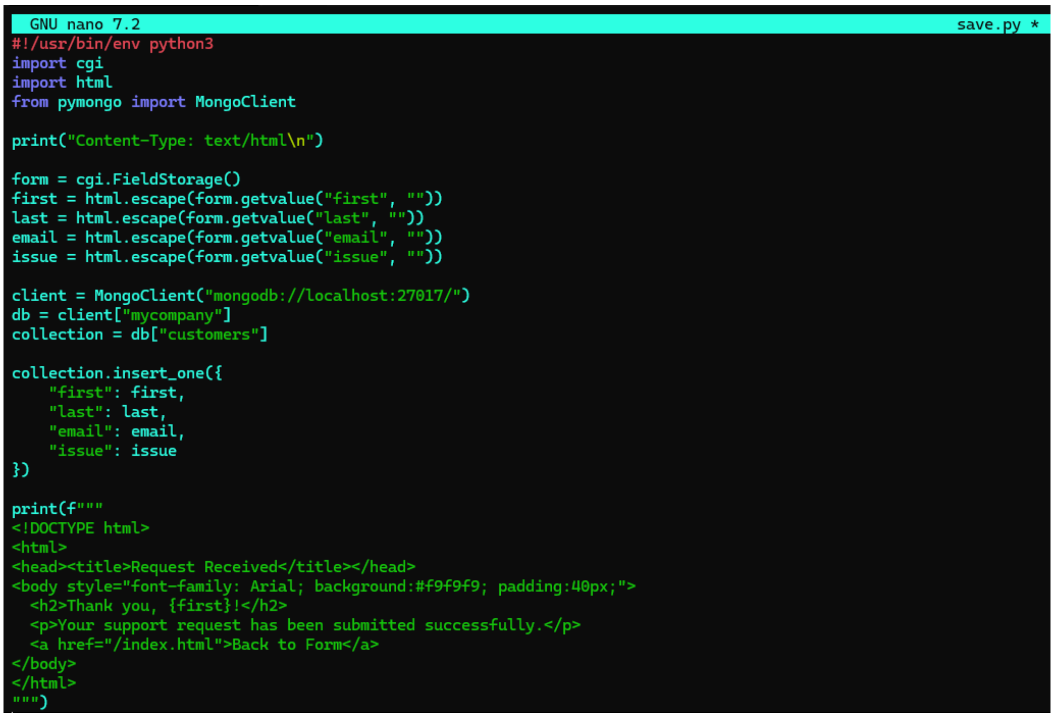
Now MongoDB was installed and ready to receive form data from the web application. To allow Apache to run Python scripts, I enabled the CGI module. See Figure 7.



```
mahmoud@ubuntu-server:~$ sudo a2enmod cgi
Your MPM seems to be threaded. Selecting cgid instead of cgi.
Enabling module cgid.
To activate the new configuration, you need to run:
  systemctl restart apache2
mahmoud@ubuntu-server:~$ sudo systemctl restart apache2
mahmoud@ubuntu-server:~$ sudo a2enmod cgi
Your MPM seems to be threaded. Selecting cgid instead of cgi.
Module cgid already enabled
mahmoud@ubuntu-server:~$
```

Figure 7: Enable CGI module

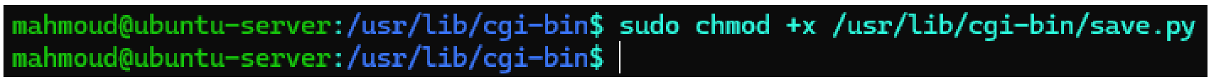
The form data is sent to a script named save.py, placed in /usr/lib/cgi-bin/. This Python script reads the data, connects to MongoDB, and saves the entry into a database called customers. See Figure 8.



```
  GNU nano 7.2                                                                          save.py *
#!/usr/bin/env python3
import cgi
import html
from pymongo import MongoClient

print("Content-Type: text/html\n")

form = cgi.FieldStorage()
first = html.escape(form.getvalue("first", ""))
last = html.escape(form.getvalue("last", ""))
email = html.escape(form.getvalue("email", ""))
issue = html.escape(form.getvalue("issue", ""))

client = MongoClient("mongodb://localhost:27017/")
db = client["mycompany"]
collection = db["customers"]

collection.insert_one({
    "first": first,
    "last": last,
    "email": email,
    "issue": issue
})

print(f"""
<!DOCTYPE html>
<html>
<head><title>Request Received</title></head>
<body style="font-family: Arial; background:#f9f9f9; padding:40px;">
  <h2>Thank you, {first}!</h2>
  <p>Your support request has been submitted successfully.</p>
  <a href="/index.html">Back to Form</a>
</body>
</html>
""")
```

Figure 8: Python script to collect customers data

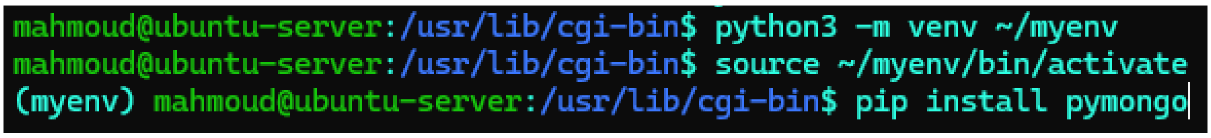
Then, I made the code executable using the command in Figure 9.



```
mahmoud@ubuntu-server:/usr/lib/cgi-bin$ sudo chmod +x /usr/lib/cgi-bin/save.py
mahmoud@ubuntu-server:/usr/lib/cgi-bin$
```

Figure 9: Make python script executable

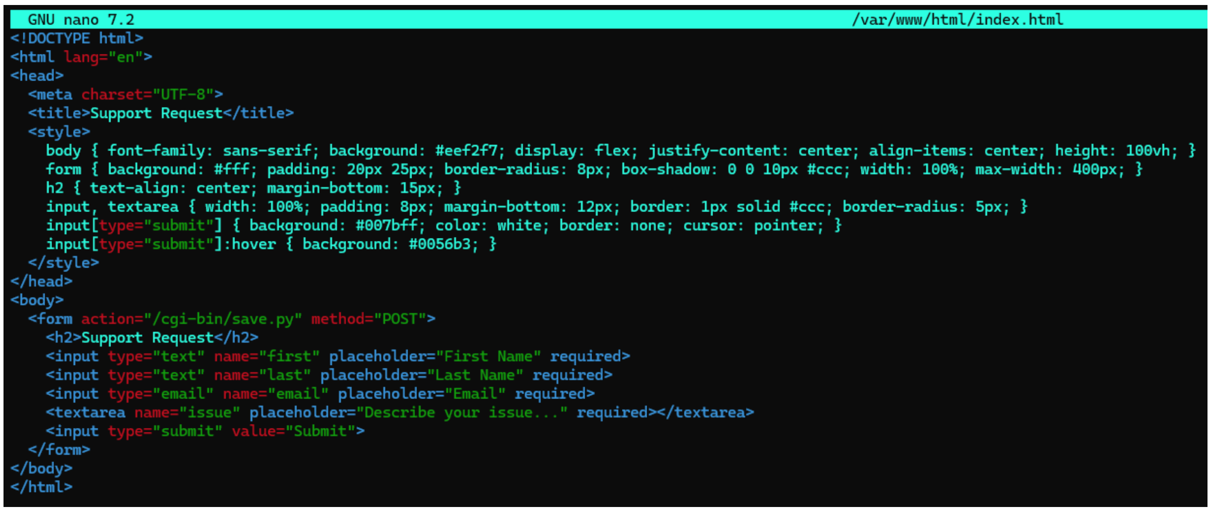
Since the system restricts direct use of pip globally, I created a virtual environment inside the CGI folder. See Figure 10.



```
mahmoud@ubuntu-server:/usr/lib/cgi-bin$ python3 -m venv ~/myenv
mahmoud@ubuntu-server:/usr/lib/cgi-bin$ source ~/myenv/bin/activate
(myenv) mahmoud@ubuntu-server:/usr/lib/cgi-bin$ pip install pymongo
```

Figure 10: Creating python virtual environment

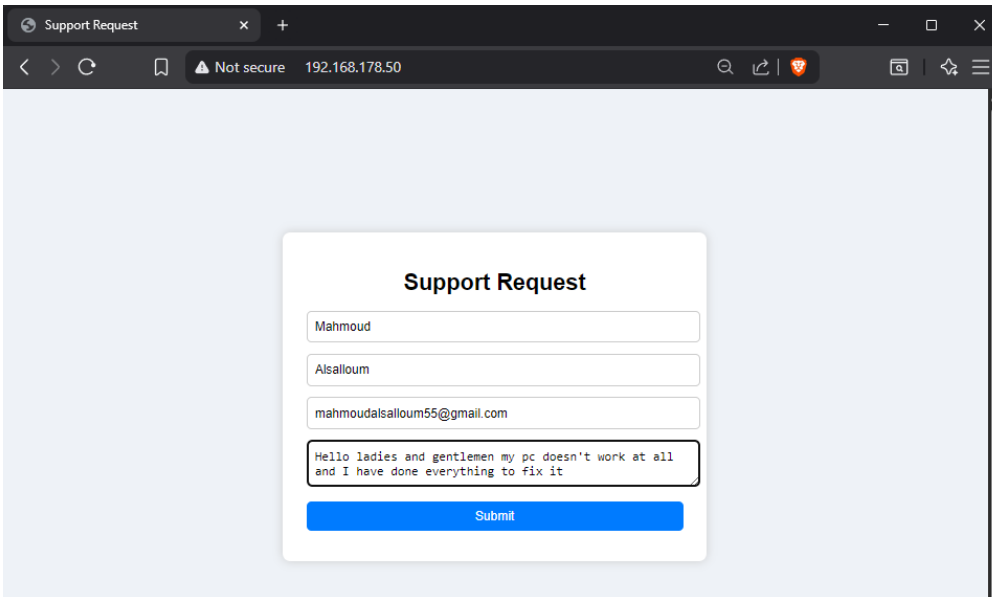
To build the interface, I created a single file named index.html. This file contains a clean and modern form built with HTML and inline CSS. It allows customers to enter their first name, last name, email address, and a description of their issue. The form uses the POST method to send data to the Python script save.py, which process and stores the information in MongoDB. Then, I stored the index.html file in Apache's web directory. See Figure 11.



```
  GNU nano 7.2                                                                          /var/www/html/index.html
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Support Request</title>
  <style>
    body { font-family: sans-serif; background: #eef2f7; display: flex; justify-content: center; align-items: center; height: 100vh; }
    form { background: #fff; padding: 20px 25px; border-radius: 8px; box-shadow: 0 0 10px #ccc; width: 100%; max-width: 400px; }
    h2 { text-align: center; margin-bottom: 15px; }
    input, textarea { width: 100%; padding: 8px; margin-bottom: 12px; border: 1px solid #ccc; border-radius: 5px; }
    input[type="submit"] { background: #007bff; color: white; border: none; cursor: pointer; }
    input[type="submit"]:hover { background: #0056b3; }
  </style>
</head>
<body>
  <form action="/cgi-bin/save.py" method="POST">
    <h2>Support Request</h2>
    <input type="text" name="first" placeholder="First Name" required>
    <input type="text" name="last" placeholder="Last Name" required>
    <input type="email" name="email" placeholder="Email" required>
    <textarea name="issue" placeholder="Describe your issue..." required></textarea>
    <input type="submit" value="Submit">
  </form>
</body>
</html>
```

Figure 11: The HTML & CSS code

To verify that everything works as expected, I opened the website in my browser by typing my IP address http://192.168.178.50. I filled out the form with test data and clicked the Submit button. After submitting, a new page appeared confirming that the data had been processed. This indicates that the HTML form successfully connected with the python CGI script and that the script executed properly. See Figure 12 and 13.



Support Request

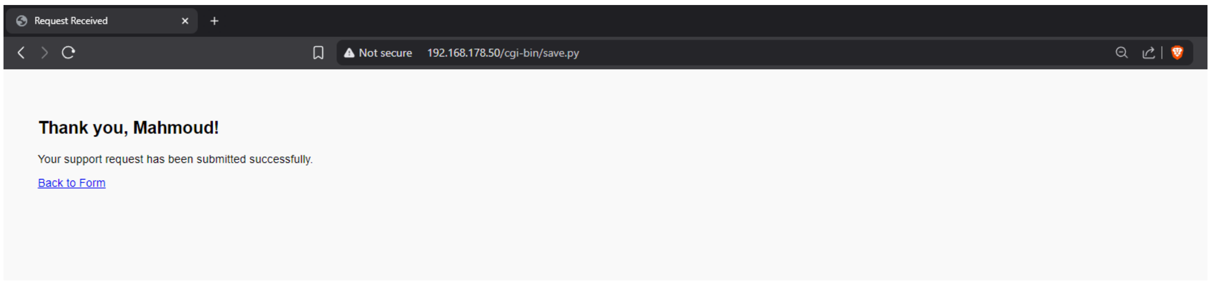
Mahmoud

Alsalloum

mahmoudalsalloum55@gmail.com

```
Hello ladies and gentlemen my pc doesn't work at all
and I have done everything to fix it
```

Figure 12: Testing the website with data entry

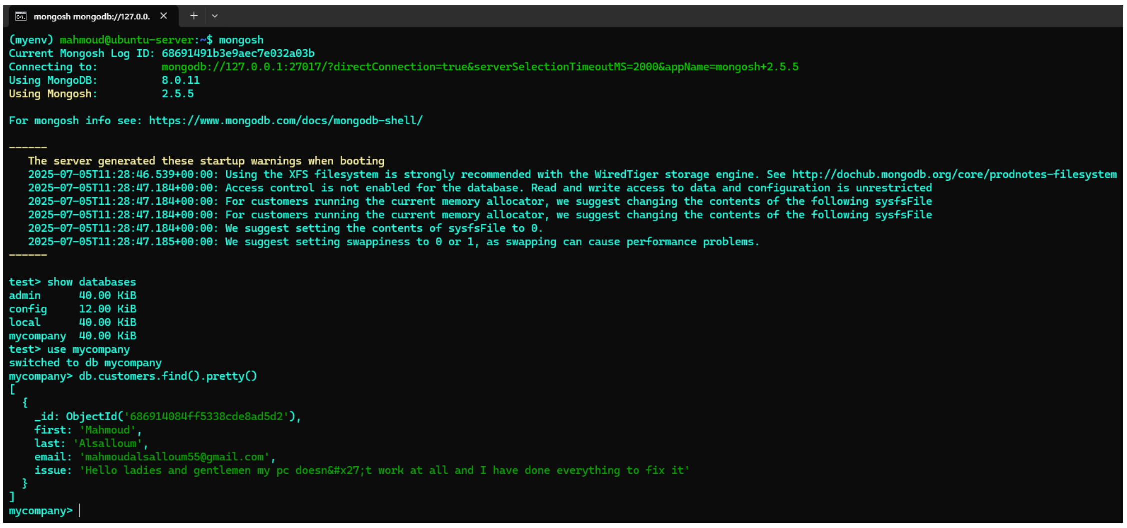


Thank you, Mahmoud!

Your support request has been submitted successfully.

Figure 13: Request verification

Later, I checked the MongoDB database to confirm that the submitted information was stored correctly. See Figure 14.



```
(myenv) mahmoud@ubuntu-server:~$ mongosh
Current Mongosh Log ID: 68691491b3e9aec7e032a03b
Connecting to:          mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000&appName=mongosh+2.5.5
Using MongoDB:          8.0.11
Using Mongosh:          2.5.5

For mongosh info see: https://www.mongodb.com/docs/mongodb-shell/

------
   The server generated these startup warnings when booting
   2025-07-05T11:28:46.539+00:00: Using the XFS filesystem is strongly recommended with the WiredTiger storage engine. See http://dochub.mongodb.org/core/prodnotes-filesystem
   2025-07-05T11:28:47.184+00:00: Access control is not enabled for the database. Read and write access to data and configuration is unrestricted
   2025-07-05T11:28:47.184+00:00: For customers running the current memory allocator, we suggest changing the contents of the following sysfsFile
   2025-07-05T11:28:47.184+00:00: For customers running the current memory allocator, we suggest changing the contents of the following sysfsFile
   2025-07-05T11:28:47.184+00:00: We suggest setting the contents of sysfsFile to 0.
   2025-07-05T11:28:47.185+00:00: We suggest setting swappiness to 0 or 1, as swapping can cause performance problems.
------

test> show databases
admin      40.00 KiB
config     12.00 KiB
local      40.00 KiB
mycompany  40.00 KiB
test> use mycompany
switched to db mycompany
mycompany> db.customers.find().pretty()
[
  {
    _id: ObjectId('686914084ff5338cde8ad5d2'),
    first: 'Mahmoud',
    last: 'Alsalloum',
    email: 'mahmoudalsalloum55@gmail.com',
    issue: 'Hello ladies and gentlemen my pc doesn&#x27;t work at all and I have done everything to fix it'
  }
]
mycompany> 
```

Figure 14: Checking database in MongoDB shell