

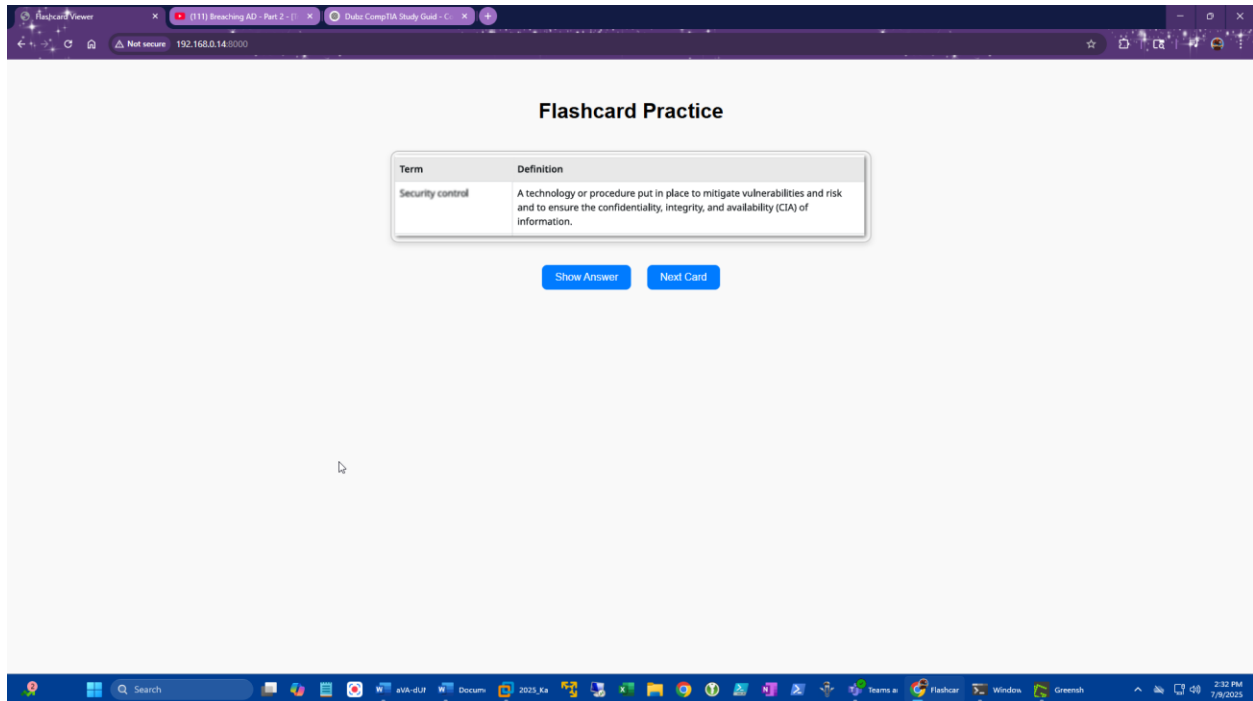
```
(dubz@kali)-[~]  
$ cd comptia-study/
```

```
(dubz@kali)-[~/comptia-study]  
$ tree
```

```
.  
├── answer1.png  
├── answer2.png  
├── answer3.png  
├── comptia-copy-dir.txt  
├── comptia-study.py  
├── index.html  
├── question1.png  
├── question2.png  
└── question3.png
```

```
1 directory, 9 files
```

```
(dubz@kali)-[~/comptia-study]  
$ █
```



## CompTIA Flashcard Web App: Step-by-Step (Ubuntu Guide)

### What You're Building

A simple offline flashcard app that:

- Shows a **question image**
- Reveals the **answer image** when you click a button
- Cycles through 3 starter questions/answers

---

### Step 1: Set Up Your Folder

1. Open your terminal.
2. Create the project directory and go into it:

```
mkdir ~/comptia-study  
cd ~/comptia-study
```

3. Add your flashcard image files:
  - o question1.png, answer1.png

- o question2.png, answer2.png
- o question3.png, answer3.png

Put them directly in ~/comptia-study

---

## ✅ Step 2: Create the Flashcard Web Page

Create a file called index.html:

```
nano index.html
```

Paste this HTML code into the file:

```
html
CopyEdit
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8" />
  <meta name="viewport" content="width=device-width, initial-scale=1.0"/>
  <title>Flashcard Practice</title>
  <style>
    body { font-family: sans-serif; text-align: center; background: #f4f4f4;
padding: 2em; }
    img { max-width: 90%; border-radius: 12px; border: 2px solid #aaa;
margin: 1em 0; }
    button { padding: 10px 20px; margin: 10px; font-size: 1em; border-radius:
6px; cursor: pointer; background: #0069ed; color: white; border: none; }
    button:hover { background: #004ba8; }
  </style>
</head>
<body>
  <h1>CompTIA Flashcard Practice</h1>
  
  <br />
  <button onclick="toggleAnswer()">Show Answer</button>
  <button onclick="nextCard()">Next Card</button>

  <script>
    const cards = [
      { question: 'question1.png', answer: 'answer1.png' },
      { question: 'question2.png', answer: 'answer2.png' },
      { question: 'question3.png', answer: 'answer3.png' },
    ];
    let currentIndex = 0;
    let showingAnswer = false;

    function toggleAnswer() {
      showingAnswer = !showingAnswer;
      document.getElementById('cardImage').src = showingAnswer
        ? cards[currentIndex].answer
```

```

        : cards[currentIndex].question;
    }

    function nextCard() {
        currentIndex = (currentIndex + 1) % cards.length;
        showingAnswer = false;
        document.getElementById('cardImage').src =
cards[currentIndex].question;
    }
</script>
</body>
</html>

```

Save the file and exit:

- Press CTRL+O → Enter to save
- Press CTRL+X to exit

---

### ✅ Step 3: Create the Python Web Server

Create a simple server file:

```
nano comptia-study.py
```

Paste this code:

```

import http.server
import socketserver
import socket

PORT = 8000
hostname = socket.gethostname()
local_ip = socket.gethostbyname(hostname)

Handler = http.server.SimpleHTTPRequestHandler

with socketserver.TCPServer(("", PORT), Handler) as httpd:
    print(f"✅ Server is running at: http://{local_ip}:{PORT}")
    print("🌐 Open it in your browser (or your phone if it's on the same Wi-Fi)")
    print("🛑 Press Ctrl+C to stop the server.")
    httpd.serve_forever()

```

Save and exit like before.

---


### ✅ Step 4: Run Your Web App

In the terminal:

```
python3 comptia-study.py
```

If your IP is 192.168.1.10, open:

```
http://192.168.1.10:8000
```

 You can open this on your **phone or tablet**, as long as it's on the **same Wi-Fi**.

---

### **Step 5: Add Your Own Questions**

Replace `question1.png` and `answer1.png` with your own CompTIA-style graphics. Use **Canva**, **PowerPoint**, or **handwritten notes** saved as PNGs.

---

### **You're All Set!**

Now you've got:

- A working **offline study tool**
- Full control to expand it with new flashcards
- A simple setup you can share with others