

Part 1 – GitHub Repo, Branching, and Flask Project

1. Create a New GitHub Repository

Command/Action:

Go to [GitHub.com](https://github.com)

Click "New" to create a repository.

Choose repository name (e.g., flask-mongo-form).

Select Public or Private as required.

Click "Create repository".

Explanation:

This step sets up a central place on GitHub where your project will be stored and version-controlled. By creating a repository, you prepare a space for your code, documentation, and any updates to be tracked and shared with others.

2. Clone Repository Using SSH

Commands:

Generate SSH Key

```
ssh-keygen -t rsa -b 4096 -C "your_email@example.com"
```

Start ssh-agent

```
eval "$(ssh-agent -s)"
```

```
ssh-add ~/.ssh/id_rsa
```

Copy Public Key to GitHub

```
cat ~/.ssh/id_rsa.pub
```

Go to GitHub → Settings → SSH and GPG keys → New SSH Key → Paste the copied key → Save.

Clone Repository

```
git clone git@github.com:username/repo_name.git
```

```
cd repo_name
```

Explanation:

SSH keys allow secure, password-free communication between your local machine and GitHub.

First, we generate a key (ssh-keygen) and add it to the SSH agent.

We then upload the public key to GitHub so GitHub can authenticate our machine.

Finally, we clone the repository using SSH, allowing us to push and pull changes securely.

3. Create a Branch with Your Username

Command:

```
git checkout -b sahal
```

Explanation:

A branch is like a separate workspace in Git. By creating a branch named after your username, you can work on your features or files without affecting the main branch. This helps in keeping your main branch clean and stable until changes are reviewed and merged.

4. Add Flask Project Files

Example app.py:

```
from flask import Flask, jsonify
```

```
app = Flask(__name__)
```

```
@app.route("/")
```

```
def home():
```

```
    return "Hello, Flask!"
```

```
@app.route("/api")
```

```
def api():
```

```
    return jsonify({"message": "API route works!"})
```

```
if __name__ == "__main__":
```

```
    app.run(debug=True)
```

Explanation:

This is a simple Flask application:

/ route → Returns a plain text message "Hello, Flask!".

/api route → Returns a JSON response {"message": "API route works!"}.

The debug=True flag allows automatic reloads and shows detailed error messages during development.

Adding these files to your branch prepares your project for deployment or further development.

5. Commit and Merge

Commands:

```
git add .  
git commit -m "Added Flask project"  
git checkout main  
git merge sahal  
git push origin main
```

Explanation:

git add . → Stages all changes for commit.

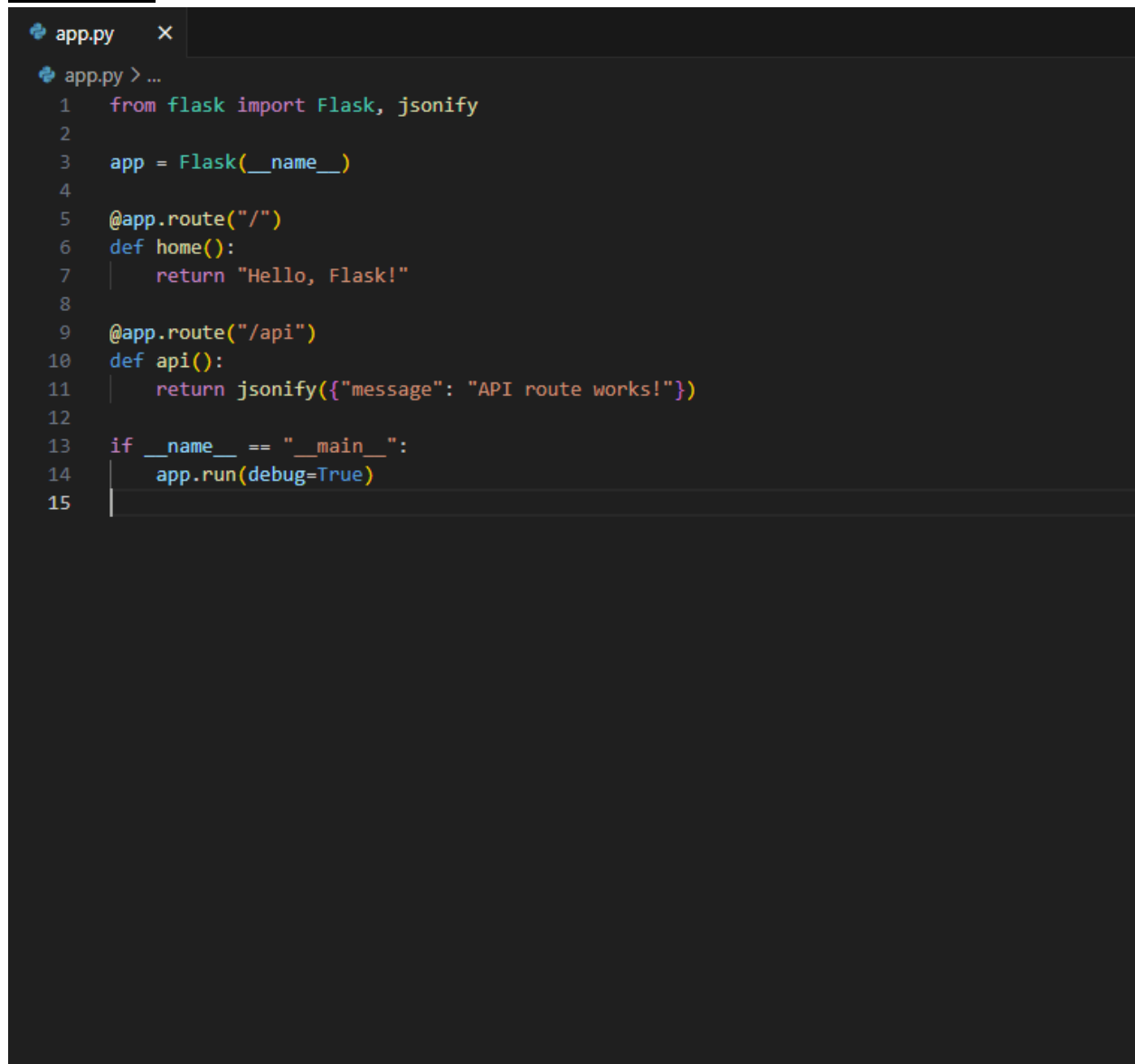
git commit -m "..." → Saves the staged changes to Git history with a message.

git checkout main → Switches to the main branch.

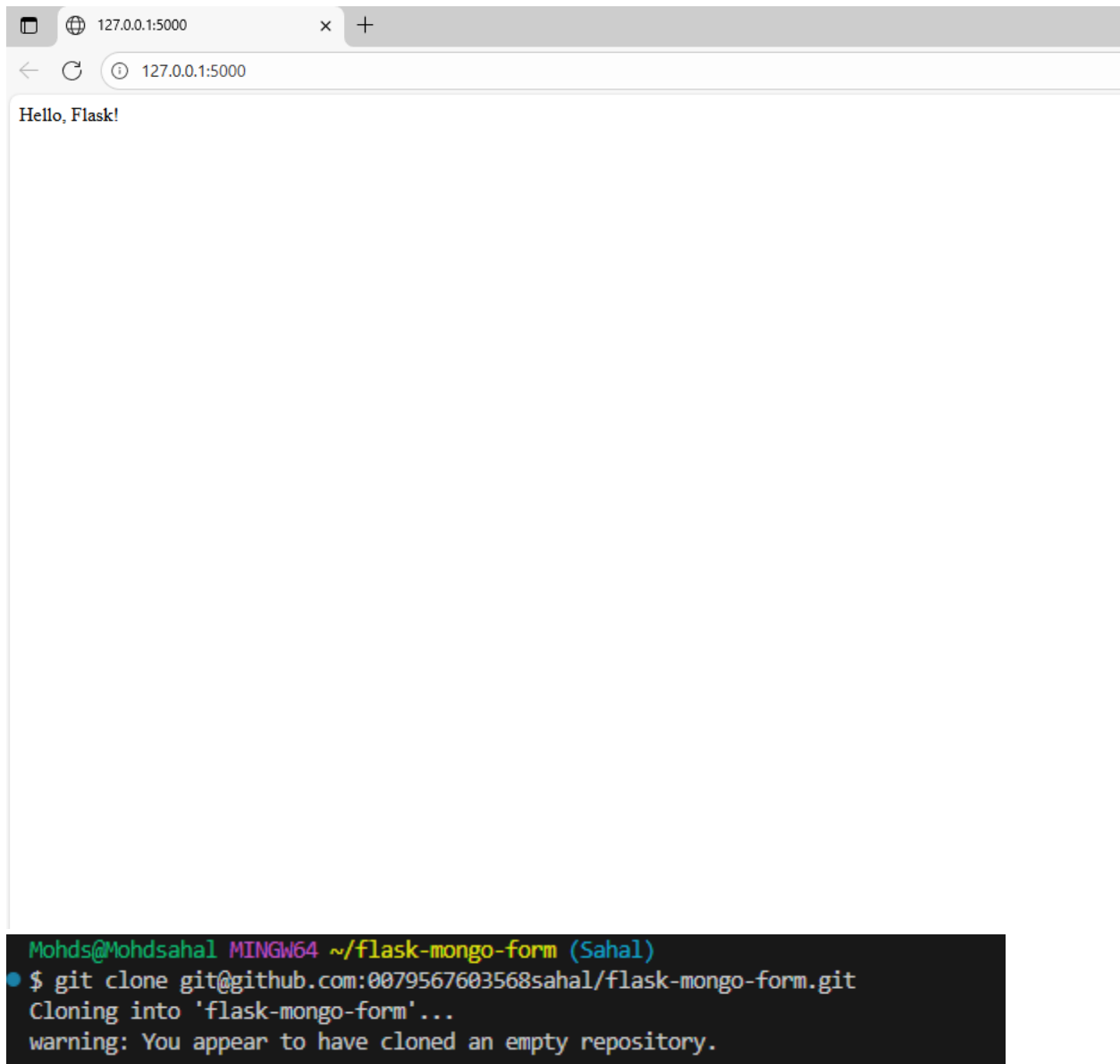
git merge sahal → Merges the work from your branch into the main branch.

git push origin main → Sends the updated main branch to GitHub.

Screenshots



```
app.py  ×
app.py > ...
1  from flask import Flask, jsonify
2
3  app = Flask(__name__)
4
5  @app.route("/")
6  def home():
7      return "Hello, Flask!"
8
9  @app.route("/api")
10 def api():
11     return jsonify({"message": "API route works!"})
12
13 if __name__ == "__main__":
14     app.run(debug=True)
15
```



```
Mohds@Mohdsahal MINGW64 ~/flask-mongo-form (sahal)
```

```
$ git checkout -b sahal  
Switched to a new branch 'sahal'
```

```
Mohds@Mohdsahal MINGW64 ~/flask-mongo-form (sahal)
```

```
$ git branch
```

```
Mohds@Mohdsahal MINGW64 ~/flask-mongo-form (sahal)
```

```
$ git checkout -b sahal  
Switched to a new branch 'sahal'
```

```
Mohds@Mohdsahal MINGW64 ~/flask-mongo-form (sahal)
```

```
$ rm -rf flask-mongo-form/.git
```

```
Mohds@Mohdsahal MINGW64 ~/flask-mongo-form (sahal)
```

```
$ git add .
```

```
Mohds@Mohdsahal MINGW64 ~/flask-mongo-form (sahal)
```

```
$ git commit -m "Initial commit with Flask project files"  
[sahal (root-commit) a06eafb] Initial commit with Flask project files  
1 file changed, 14 insertions(+)  
create mode 100644 app.py
```

```
Mohds@Mohdsahal MINGW64 ~/flask-mongo-form (sahal)
```

```
$ git push origin sahal  
Enumerating objects: 3, done.  
Counting objects: 100% (3/3), done.  
Delta compression using up to 12 threads  
Compressing objects: 100% (2/2), done.  
Writing objects: 100% (3/3), 385 bytes | 385.00 KiB/s, done.  
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)  
To github.com:0079567603568sahal/flask-mongo-form.git  
* [new branch] sahal -> sahal
```

```
Mohds@Mohdsahal MINGW64 ~/flask-mongo-form (sahal)
```

```
$ # Switch to main branch (create it if it doesn't exist yet)  
git checkout -b main
```

```
# Merge sahal branch into main  
git merge sahal
```

```
# Push main branch to GitHub  
git push origin main  
Switched to a new branch 'main'  
Your branch is based on 'origin/main', but the upstream is gone.  
(use "git branch --unset-upstream" to fixup)
```

```
(use "git branch --unset-upstream" to fixup)
Already up to date.
Total 0 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
remote:
remote: Create a pull request for 'main' on GitHub by visiting:
remote:   https://github.com/0079567603568sahal/flask-mongo-form/pull/new/main
remote:
To github.com:0079567603568sahal/flask-mongo-form.git
* [new branch]      main -> main
```

```
$ git add .
git commit -m "Added Flask project"
git checkout main
git merge sahal
git push origin main
On branch main
Your branch is up to date with 'origin/main'.

nothing to commit, working tree clean
Already on 'main'
Your branch is up to date with 'origin/main'.
Already up to date.
Everything up-to-date
```

Part 2 – JSON Update and Conflict Resolution

1. Create a new branch named sahal_new

```
git checkout -b sahal_new
```

Explanation:

This command creates a new branch called sahal_new and switches to it immediately. Branches allow you to work on changes without affecting the main branch.

2. Edit data.json (used in /api route)

```
{  
  "message": "Updated content for Sahal"  
}
```

Explanation:

Here, the data.json file is modified to update its message. This change will be committed to the new branch. The /api route in your Flask app likely reads this file to display updated content.

3. Stage the modified file for commit

```
git add data.json
```

Explanation:

This stages the updated data.json file, telling Git to include it in the next commit.

4. Commit the changes

```
git commit -m "Updated JSON file in sahal_new"
```

Explanation:

This creates a commit in the sahal_new branch with a message describing the change.

5. Switch back to the main branch

```
git checkout main
```

Explanation:

Changes will now be merged into the main branch, so we switch back from sahal_new to main.

6. Merge sahal_new into main

```
git merge sahal_new
```

Explanation:

This merges the changes from sahal_new into main. If there are no conflicts, the merge completes automatically.

7. If there is a merge conflict

```
# Open conflicted file → keep sahal_new changes → save  
git add data.json  
git commit -m "Resolved merge conflict using sahal_new version"
```

Explanation:

If Git detects conflicting changes in data.json, it will ask you to resolve them manually.

Open the file, choose the sahal_new changes, and save it.

Stage and commit the resolved file.

8. Push changes to remote repository

```
git push origin main
```

Explanation:

This uploads your updated main branch to the remote repository (e.g., GitHub).

Part 3 – Feature Development

1. Ensure you are on the main branch

```
git checkout main
```

Explanation:

Always start new features from the latest main branch to avoid outdated code.

2. Create a new branch master_1

```
git checkout -b master_1
```

Explanation:

Creates and switches to a new branch master_1, where you can work on a specific feature independently.

3. Switch back to main

```
git checkout main
```

Explanation:

Return to the main branch to start another feature branch.

4. Create another branch master_2

```
git checkout -b master_2
```

Explanation:

Creates and switches to another feature branch named master_2

Screenshots

app.py

data.json 1

C: > Users > Mohds > Desktop > git assignmrnt > {} data.json

```
1  git add data.json
2  git commit -m "Updated JSON file in sahal_new"
3  git checkout main
4  git merge sahal_new
5  # If conflict:
6  # Open conflicted file → keep sahal_new changes → save
7  git add data.json
8  git commit -m "Resolved merge conflict using sahal_new version"
9  git push origin main
10 {}
11 "message": "Updated content for Sahal"
12 }
13
```

PROBLEMS 1 OUTPUT DEBUG CONSOLE TERMINAL PORTS

Mohds@Mohdsahal MINGW64 ~/flask-mongo-form (main)

```
• $ git checkout -b sahal_new
Switched to a new branch 'sahal_new'
```

Mohds@Mohdsahal MINGW64 ~/flask-mongo-form (sahal_new)

```
$ git add data.json
git commit -m "Updated JSON file in sahal_new"
git checkout main
git merge sahal_new
# If conflict:
# Open conflicted file → keep sahal_new changes → save
• git add data.json
```

```
• git add data.json
git commit -m "Resolved merge conflict using sahal_new version"
git push origin main
fatal: pathspec 'data.json' did not match any files
On branch sahal_new
nothing to commit, working tree clean
Switched to branch 'main'
Your branch is up to date with 'origin/main'.
Already up to date.
fatal: pathspec 'data.json' did not match any files
On branch main
Your branch is up to date with 'origin/main'.
```

```
nothing to commit, working tree clean
Switched to branch 'main'
Your branch is up to date with 'origin/main'.
Already up to date.
fatal: pathspec 'data.json' did not match any files
On branch main
Your branch is up to date with 'origin/main'.

nothing to commit, working tree clean
Everything up-to-date
```

Part 3 – Feature Development

App.py

Import required libraries

from flask import Flask, render_template, request, redirect, url_for

from pymongo import MongoClient

import certifi

import os

Load MongoDB URI from environment variable

MONGO_URI = os.getenv(

"MONGO_URI",

"mongodb+srv://sahalusr1:securepass@cluster0.bdnymrmo.mongodb.net/?retryWrites=true&w=majority&appName=Cluster0"

)

Create the Flask app

app = Flask(__name__)

Route to handle form submission

@app.route("/submittodoitem", methods=["POST"])

def submit_todo_item():

"""

Handles form submission from your HTML form.

Inserts the submitted itemName and itemDescription into MongoDB.

```
"""
```

```
# Get the submitted data from the form
```

```
item_name = request.form.get("itemName")
```

```
item_description = request.form.get("itemDescription")
```

```
if item_name and item_description:
```

```
    # Connect to MongoDB
```

```
    client = MongoClient(MONGO_URI, tlsCAFile=certifi.where())
```

```
    db = client["todo_database"]
```

```
    collection = db["todos"]
```

```
    # Insert the submitted item into MongoDB
```

```
    collection.insert_one({
```

```
        "name": item_name,
```

```
        "description": item_description
```

```
    })
```

```
# Redirect back to the home page showing all tasks
```

```
return redirect(url_for("show_todo_list"))
```

```
# Route to display the todo list
```

```
@app.route("/")
```

```
def show_todo_list():
```

```
    # Connect to MongoDB
```

```
    client = MongoClient(MONGO_URI, tlsCAFile=certifi.where())
```

```
    db = client["todo_database"]
```

```
    collection = db["todos"]
```

```
# Fetch all tasks

tasks = collection.find()

# Render tasks to HTML template

return render_template("todo.html", tasks=tasks)

# Run the Flask app

if __name__ == "__main__":

    app.run(debug=True)
```

Todo.html

```
<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <title>Todo List</title>

    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.min.css"
rel="stylesheet">

</head>

<body class="container mt-5">

    <h1 class="mb-4">Todo List</h1>

    <!-- Form to add new todo -->

    <form action="{{ url_for('submit_todo_item') }}" method="post" class="mb-4">

        <div class="mb-3">

            <input type="text" name="itemName" class="form-control" placeholder="Item Name"
required>

        </div>

        <div class="mb-3">
```

```

        <textarea name="itemDescription" class="form-control" placeholder="Item Description"
required></textarea>

    </div>

    <button type="submit" class="btn btn-primary">Submit</button>

</form>

<!-- Display todo list -->
<ul class="list-group">
    {% for task in tasks %}
        <li class="list-group-item">
            <strong>{{ task.name }}</strong>: {{ task.description }}
        </li>
    {% endfor %}
</ul>

</body>
</html>

```

Explanation

we create two separate branches for different features of the To-Do application:

master_1 → Frontend (HTML Form)

master_2 → Backend (MongoDB Integration)

This approach allows parallel development without interfering with each other's work.

Step 1 – Create Feature Branches

```
git checkout main
```

```
git checkout -b master_1
```

```
git checkout main
```

```
git checkout -b master_2
```

Explanation:

Switch to the main branch to ensure you start from the latest code.

Create a new branch master_1 for the frontend work.

Switch back to main again.

Create another branch master_2 for the backend work.

Step 2 – Develop master_1 (To-Do Frontend)

File: templates/todo.html

```
<form action="/submittodoitem" method="post">
  <input type="text" name="itemName" placeholder="Item Name"><br>
  <textarea name="itemDescription" placeholder="Item Description"></textarea><br>
  <button type="submit">Submit</button>
</form>
```

Explanation:

This HTML form collects item name and description from the user.

When submitted, it sends data to the /submittodoitem route using the POST method.

The form will later be processed by the backend in master_2.

Git Commands to Commit and Push:

```
git add templates/todo.html
git commit -m "Added To-Do page frontend"
git push origin master_1
```

Explanation:

git add stages the new todo.html file.

git commit saves the changes locally with a meaningful message.

git push uploads the branch to GitHub.

Step 3 – Develop master_2 (Backend MongoDB)

Code Update in app.py:

```
from flask import request
from pymongo import MongoClient
client = MongoClient("mongodb://localhost:27017/")
db = client.todoDB
@app.route("/submittodoitem", methods=["POST"])
def submit_todo():
    name = request.form["itemName"]
```



```
desc = request.form["itemDescription"]
db.todos.insert_one({"name": name, "description": desc})
return "Item Added!"
```

Explanation:

Connects to a local MongoDB database (todoDB).

Defines a /submittodoitem route that accepts POST requests.

Reads itemName and itemDescription from the form.

Inserts them into the todos collection in MongoDB.

Returns a confirmation message "Item Added!".

Git Commands to Commit and Push:

```
git add app.py
git commit -m "Added backend route to store To-Do item in MongoDB"
git push origin master_2
```

Explanation:

Stages the updated app.py.

Commits with a descriptive message.

Pushes master_2 branch to GitHub.

Step 4 – Merge Both Features into Main

```
git checkout main
git merge master_1
git merge master_2
git push origin main
```

Explanation:

Switch back to main.

Merge master_1 (frontend) changes into main.

Merge master_2 (backend) changes into main.

Push the updated main branch to GitHub so it contains both frontend and backend functionality.

Screenshots

```

app.py M X  < todo.html M
app.py > ...
1  # Import required libraries
2  from flask import Flask, render_template, request, redirect, url_for
3  from pymongo import MongoClient
4  import certifi
5  import os
6
7  # Load MongoDB URI from environment variable
8  MONGO_URI = os.getenv(
9      "MONGO_URI",
10     "mongodb+srv://sahalusr1:securepass@cluster0.bdnymr.mongodb.net/?retryWrites=true&w=majority&appName=Cluster0"
11 )
12
13 # Create the Flask app
14 app = Flask(__name__)
15
16 # Route to handle form submission
17 @app.route("/submittodoitem", methods=["POST"])
18 def submit_todo_item():
19     """
20     Handles form submission from your HTML form.
21     Inserts the submitted itemName and itemDescription into MongoDB.
22     """
23     # Get the submitted data from the form
24     item_name = request.form.get("itemName")
25     item_description = request.form.get("itemDescription")
26
27     if item_name and item_description:
28         # Connect to MongoDB
29         client = MongoClient(MONGO_URI, tlsCAFile=certifi.where())
30         db = client["todo_database"]
31         collection = db["todos"]
32
33         # Insert the submitted item into MongoDB
34         collection.insert_one({

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```

* Detected change in 'c:\Users\Wohds\flask-mongo-form\app.py', reloading
* Restarting with stat
* Debugger is active!
* Debugger PIN: 375-489-153
127.0.0.1 - - [25/Aug/2025 15:26:42] "GET / HTTP/1.1" 200 -
127.0.0.1 - - [25/Aug/2025 15:26:49] "POST /submittodoitem HTTP/1.1" 302 -
127.0.0.1 - - [25/Aug/2025 15:26:50] "GET / HTTP/1.1" 200 -
127.0.0.1 - - [25/Aug/2025 15:28:00] "GET / HTTP/1.1" 200 -
127.0.0.1 - - [25/Aug/2025 15:28:13] "POST /submittodoitem HTTP/1.1" 302 -
127.0.0.1 - - [25/Aug/2025 15:28:14] "GET / HTTP/1.1" 200 -
127.0.0.1 - - [25/Aug/2025 15:28:36] "POST /submittodoitem HTTP/1.1" 302 -
127.0.0.1 - - [25/Aug/2025 15:28:37] "GET / HTTP/1.1" 200 -

```

```
app.py M  todo.html M X
templates > todo.html > html > body.container.mt-5
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4    <meta charset="UTF-8">
5    <title>Todo List</title>
6    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.min.css" rel="stylesheet">
7  </head>
8  <body class="container mt-5">
9
10   <h1 class="mb-4">Todo List</h1>
11
12   <!-- Form to add new todo -->
13   <form action="{{ url_for('submit_todo_item') }}" method="post" class="mb-4">
14     <div class="mb-3">
15       <input type="text" name="itemName" class="form-control" placeholder="Item Name" required>
16     </div>
17     <div class="mb-3">
18       <textarea name="itemDescription" class="form-control" placeholder="Item Description" required></textarea>
19     </div>
20     <button type="submit" class="btn btn-primary">Submit</button>
21   </form>
22
23   <!-- Display todo list -->
24   <ul class="list-group">
25     {% for task in tasks %}
26     <li class="list-group-item">
27       <strong>{{ task.name }}</strong>: {{ task.description }}
28     </li>
29     {% endfor %}
30   </ul>
31
32 </body>
33 </html>
34
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS
* Detected change in 'c:\Users\Wohds\flask-mongo-form\app.py', reloading
* Restarting with stat
* Debugger is active!
* Debugger PIN: 375-489-153
127.0.0.1 - - [25/Aug/2025 15:26:42] "GET / HTTP/1.1" 200 -
127.0.0.1 - - [25/Aug/2025 15:26:49] "POST /submittodoitem HTTP/1.1" 302 -
127.0.0.1 - - [25/Aug/2025 15:26:50] "GET / HTTP/1.1" 200 -
127.0.0.1 - - [25/Aug/2025 15:28:00] "GET / HTTP/1.1" 200 -
127.0.0.1 - - [25/Aug/2025 15:28:13] "POST /submittodoitem HTTP/1.1" 302 -
127.0.0.1 - - [25/Aug/2025 15:28:14] "GET / HTTP/1.1" 200 -
127.0.0.1 - - [25/Aug/2025 15:28:36] "POST /submittodoitem HTTP/1.1" 302 -
127.0.0.1 - - [25/Aug/2025 15:28:37] "GET / HTTP/1.1" 200 -

```

127.0.0.1:5000

Todo List

zxzx: zxzxzx

orange: dfgdfg

apple: apple is red

Part 4 – Enhancements + Git Reset + Rebase

```
from flask import Flask, render_template, request, redirect, url_for
from pymongo import MongoClient
import os
```

Load MongoDB URI from environment variable or use default

```
MONGO_URI = os.getenv( "MONGO_URI",
"mongodb+srv://sahalusr1:securepass@cluster0.bdnymrmo.mongodb.net/?retryWrites=true&w=majority&appName=Cluster0" )
```

Initialize Flask app

```
app = Flask(name)
```

Initialize MongoDB client and collection

```
client = MongoClient(MONGO_URI, tlsCAFile=certifi.where()) db = client["todo_database"]
collection = db["todos"]
```

Route to display the todo list

```
@app.route("/") def index(): tasks = list(collection.find()) return render_template("to_do.html",
tasks=tasks)
```

Route to handle form submission

```
@app.route("/submittodoitem", methods=["POST"]) def submit_todo_item(): # Collect all fields
from form itemID = request.form.get("itemID") itemUUID = request.form.get("itemUUID")
itemHash = request.form.get("itemHash") itemName = request.form.get("itemName")
itemDescription = request.form.get("itemDescription")
```

if itemID and itemUUID and itemHash and itemName and itemDescription:

```
    # Insert into MongoDB
    collection.insert_one({
        "itemID": itemID,
        "itemUUID": itemUUID,
        "itemHash": itemHash,
        "name": itemName,
        "description": itemDescription
    })
```

```
# Redirect back to the todo list page
return redirect(url_for("index"))
```

Run the Flask app

```
if name == "main": app.run(debug=True)
```

Todo.html

```
<!DOCTYPE html>
```

```
<html lang="en">
```

```
<head>
```

```
    <meta charset="UTF-8">
```

```
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
```

```
    <title>Todo List</title>
```

```
    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.min.css"
rel="stylesheet">
```

```
</head>
```

```
<body class="container mt-5">
```

```
    <h1 class="mb-4 text-center">Todo List</h1>
```

```
    <!-- Form to add new todo -->
```

```
    <form action="{{ url_for('submit_todo_item') }}" method="post" class="mb-5">
```

```
        <div class="row g-3">
```

```
            <div class="col-md-4">
```

```
                <input type="text" name="itemID" class="form-control" placeholder="Item ID"
required>
```

```
            </div>
```

```
            <div class="col-md-4">
```

```
                <input type="text" name="itemUUID" class="form-control" placeholder="Item
UUID" required>
```

```
            </div>
```

```
            <div class="col-md-4">
```

```

        <input type="text" name="itemHash" class="form-control" placeholder="Item Hash"
required>
    </div>
</div>
<div class="mt-3">
    <input type="text" name="itemName" class="form-control mb-3" placeholder="Item
Name" required>
    <textarea name="itemDescription" class="form-control" placeholder="Item Description"
rows="3" required></textarea>
</div>
    <button type="submit" class="btn btn-primary mt-3">Submit</button>
</form>

<!-- Display todo list -->
<ul class="list-group">
    {% for task in tasks %}
        <li class="list-group-item">
            <div class="mb-2">
                <span class="badge bg-secondary">ID: {{ task.itemID }}</span>
                <span class="badge bg-info text-dark">UUID: {{ task.itemUUID }}</span>
                <span class="badge bg-warning text-dark">Hash: {{ task.itemHash }}</span>
            </div>
            <strong>{{ task.name }}</strong>: {{ task.description }}
        </li>
    {% endfor %}
</ul>
</body>
</html>

```

Enhancements in master_1

```
git checkout master_1
# 1st commit - Add Item ID field
# (Edit todo.html)
<input type="text" name="itemID" placeholder="Item ID">
git add templates/todo.html
git commit -m "Added Item ID field"

# 2nd commit - Add Item UUID field
<input type="text" name="itemUUID" placeholder="Item UUID">
git add templates/todo.html
git commit -m "Added Item UUID field"

# 3rd commit - Add Item Hash field
<input type="text" name="itemHash" placeholder="Item Hash">
git add templates/todo.html
git commit -m "Added Item Hash field"
git push origin master_1

Merge into main

git checkout main
git merge master_1

Git Reset to first commit (Item ID only)

git log # find commit hash for "Added Item ID field"
git reset --soft <commit_hash>
git commit -m "Revert to only Item ID field"
git push origin main --force

Rebase changes

git checkout master_1
git rebase main
git push origin master_1 --force
```

Explanation

1. Enhancements in master_1

```
git checkout master_1
```

Switches to the master_1 branch where we will make frontend enhancements.

First commit – Add Item ID field

```
<input type="text" name="itemID" placeholder="Item ID">
```

```
git add templates/todo.html
```

```
git commit -m "Added Item ID field"
```

A new input field for Item ID is added to the To-Do page.

git add stages the change, and git commit saves it as a snapshot in the branch history.

Second commit – Add Item UUID field

```
<input type="text" name="itemUUID" placeholder="Item UUID">
```

```
git add templates/todo.html
```

```
git commit -m "Added Item UUID field"
```

Another input field Item UUID is added to the same HTML file.

This is committed as a separate commit to keep each enhancement logically independent.

Third commit – Add Item Hash field

```
<input type="text" name="itemHash" placeholder="Item Hash">
```

```
git add templates/todo.html
```

```
git commit -m "Added Item Hash field"
```

A third enhancement: Item Hash input field.

Separate commits help in tracing changes and make future debugging easier.

```
git push origin master_1
```

Pushes all three commits from master_1 to the remote repository.

2. Merge into main

```
git checkout main
```

```
git merge master_1
```

Switches to the main branch and merges changes from master_1.

Now main has all enhancements: Item ID, UUID, and Hash.

3. Git Reset to first commit (Item ID only)

```
git log # find commit hash for "Added Item ID field"
git reset --soft <commit_hash>
git commit -m "Revert to only Item ID field"
git push origin main --force
```

Purpose: Undo all commits after the first one while keeping changes staged (--soft).

Creates a new commit representing only the first enhancement (Item ID).

--force is used because the branch history is rewritten; it overwrites the remote history with this new state.

4. Rebase changes in master_1

```
git checkout master_1
git rebase main
git push origin master_1 --force
```

Rebases master_1 onto the current main.

Ensures that master_1 incorporates the latest state of main (after reset) cleanly.

--force pushes the rebased branch since its history has been rewritten.

After this, master_1 is synchronized with main but still retains its original commits, now applied on top of main.

Commands

```
Mohds@Mohdsahal MINGW64 ~/flask-mongo-form (master_2) $ cd C:\Users\Mohds\flask-mongo-form
git checkout master_1
bash: cd: C:\Users\Mohds\flask-mongo-form: No such file or directory
error: pathspec 'master_1' did not match any file(s) known to git
```

```
Mohds@Mohdsahal MINGW64 ~/flask-mongo-form (master_2) $ cd /c/Users/Mohds/flask-mongo-form
```

```
Mohds@Mohdsahal MINGW64 ~/flask-mongo-form (master_2) $ git branch
```

```
Mohds@Mohdsahal MINGW64 ~/flask-mongo-form (master_2) $ git checkout main
git pull origin main # make sure main is up to date
git checkout -b master_1
error: pathspec 'main' did not match any file(s) known to git
fatal: 'origin' does not appear to be a git repository
fatal: Could not read from remote repository.
```

Please make sure you have the correct access rights and the repository exists. Switched to a new branch 'master_1'

```
Mohds@Mohdsahal MINGW64 ~/flask-mongo-form (master_1) $ templates/to_do.html
bash: templates/to_do.html: No such file or directory
```

Mohds@Mohdsahal MINGW64 ~/flask-mongo-form (master_1) \$ ls to_do.html todo.html.txt

Mohds@Mohdsahal MINGW64 ~/flask-mongo-form (master_1) \$ ls templates

Mohds@Mohdsahal MINGW64 ~/flask-mongo-form (master_1) \$ code templates/to_do.html

Mohds@Mohdsahal MINGW64 ~/flask-mongo-form (master_1) \$ # First, stage only Item ID changes
git add templates/to_do.html git commit -m "Add Item ID field to To-Do form"

Then, stage Item UUID changes

git add templates/to_do.html git commit -m "Add Item UUID field to To-Do form"

Then, stage Item Hash changes

```
git add templates/to_do.html git commit -m "Add Item Hash field to To-Do form" [master_1
(root-commit) ea4cec8] Add Item ID field to To-Do form 1 file changed, 51 insertions(+) create
mode 100644 flask-mongo-form/templates/to_do.html warning: could not open directory
'AppData/Local/Application Data/': Permission denied warning: could not open directory
'Application Data/': Permission denied warning: could not open directory 'Cookies/': Permission
denied warning: could not open directory 'Local Settings/': Permission denied warning: could not
open directory 'My Documents/': Permission denied warning: could not open directory
'NetHood/': Permission denied warning: could not open directory 'PrintHood/': Permission denied
warning: could not open directory 'Recent/': Permission denied warning: could not open
directory 'SendTo/': Permission denied warning: could not open directory 'Start Menu/':
Permission denied warning: could not open directory 'Templates/': Permission denied On branch
master_1 Untracked files: (use "git add ..." to include in what will be
committed) ../anaconda/ ../conda/ ../condarc ../continuum/ ../gitconfig ../ipython/ ../jupyter/ ..
../keras/ ../lessht ../matplotlib/ ../spyder-
py3/ ../ssh/ ../vscode/ ../AppData/ ../Contacts/ ../Desktop/ ../Documents/ ../Downloads/ ../Favorit
es/ ../Intel/ ../Links/ ../Music/ ../NTUSER.DAT ../NTUSER.DAT{3e3a1365-569d-11f0-9a0c-
3887d58b62e8}.TM.blf ../NTUSER.DAT{3e3a1365-569d-11f0-9a0c-
3887d58b62e8}.TMContainer00000000000000000001.regtrans-
ms ../NTUSER.DAT{3e3a1365-569d-11f0-9a0c-
3887d58b62e8}.TMContainer00000000000000000002.regtrans-
ms ../NTUSER.DAT{e35b560e-4c28-11f0-99cd-3887d58b62e4}.TxR.0.regtrans-
ms ../NTUSER.DAT{e35b560e-4c28-11f0-99cd-3887d58b62e4}.TxR.1.regtrans-
ms ../NTUSER.DAT{e35b560e-4c28-11f0-99cd-3887d58b62e4}.TxR.2.regtrans-
ms ../NTUSER.DAT{e35b560e-4c28-11f0-99cd-
3887d58b62e4}.TxR.blf ../NTUSER.DAT{e35b560f-4c28-11f0-99cd-
3887d58b62e4}.TM.blf ../NTUSER.DAT{e35b560f-4c28-11f0-99cd-
3887d58b62e4}.TMContainer00000000000000000001.regtrans-ms ../NTUSER.DAT{e35b560f-
4c28-11f0-99cd-3887d58b62e4}.TMContainer00000000000000000002.regtrans-
ms ../OneDrive/ ../Pictures/ ../Saved
Games/ ../Searches/ ../Videos/ ../anaconda3/ ../day4_youtube_clone/ to_do.html
```

```
todo.html.txt ../form.html ../miniconda3/ ../ntuser.dat.LOG1 ../ntuser.dat.LOG2 ../ntuser.ini ../php-docs-hello-world/
```

```
nothing added to commit but untracked files present (use "git add" to track) warning: could not open directory 'AppData/Local/Application Data/': Permission denied warning: could not open directory 'Application Data/': Permission denied warning: could not open directory 'Cookies/': Permission denied warning: could not open directory 'Local Settings/': Permission denied warning: could not open directory 'My Documents/': Permission denied warning: could not open directory 'NetHood/': Permission denied warning: could not open directory 'PrintHood/': Permission denied warning: could not open directory 'Recent/': Permission denied warning: could not open directory 'SendTo/': Permission denied warning: could not open directory 'Start Menu/': Permission denied warning: could not open directory 'Templates/': Permission denied On branch master_1 Untracked files: (use "git add ..." to include in what will be committed) ../.anaconda/ ../.conda/ ../.condarc ../.continuum/ ../.gitconfig ../.ipython/ ../jupyter/ ../.keras/ ../.lessht ../.matplotlib/ ../.spyder-py3/ ../.ssh/ ../.vscode/ ../AppData/ ../Contacts/ ../Desktop/ ../Documents/ ../Downloads/ ../Favorites/ ../Intel/ ../Links/ ../Music/ ../NTUSER.DAT ../NTUSER.DAT{3e3a1365-569d-11f0-9a0c-3887d58b62e8}.TM.blf ../NTUSER.DAT{3e3a1365-569d-11f0-9a0c-3887d58b62e8}.TM.Container00000000000000000001.regtrans-ms ../NTUSER.DAT{3e3a1365-569d-11f0-9a0c-3887d58b62e8}.TM.Container00000000000000000002.regtrans-ms ../NTUSER.DAT{e35b560e-4c28-11f0-99cd-3887d58b62e4}.TxR.0.regtrans-ms ../NTUSER.DAT{e35b560e-4c28-11f0-99cd-3887d58b62e4}.TxR.1.regtrans-ms ../NTUSER.DAT{e35b560e-4c28-11f0-99cd-3887d58b62e4}.TxR.2.regtrans-ms ../NTUSER.DAT{e35b560e-4c28-11f0-99cd-3887d58b62e4}.TxR.blf ../NTUSER.DAT{e35b560f-4c28-11f0-99cd-3887d58b62e4}.TM.blf ../NTUSER.DAT{e35b560f-4c28-11f0-99cd-3887d58b62e4}.TM.Container00000000000000000001.regtrans-ms ../NTUSER.DAT{e35b560f-4c28-11f0-99cd-3887d58b62e4}.TM.Container00000000000000000002.regtrans-ms ../OneDrive/ ../Pictures/ ../Saved Games/ ../Searches/ ../Videos/ ../anaconda3/ ../day4_youtube_clone/ to_do.html todo.html.txt ../form.html ../miniconda3/ ../ntuser.dat.LOG1 ../ntuser.dat.LOG2 ../ntuser.ini ../php-docs-hello-world/
```

```
nothing added to commit but untracked files present (use "git add" to track)
```

```
Mohds@Mohdsahal MINGW64 ~/flask-mongo-form (master_1) $ cd /c/Users/Mohds/flask-mongo-form
```

```
Mohds@Mohdsahal MINGW64 ~/flask-mongo-form (master_1) $ rm -rf .git # careful: deletes the current Git repo git init Initialized empty Git repository in C:/Users/Mohds/flask-mongo-form/.git/
```

Mohds@Mohdsahal MINGW64 ~/flask-mongo-form (master) \$ git add templates/to_do.html git commit -m "Add Item ID field to To-Do form" [master (root-commit) 86b7ba4] Add Item ID field to To-Do form 1 file changed, 51 insertions(+) create mode 100644 templates/to_do.html

Mohds@Mohdsahal MINGW64 ~/flask-mongo-form (master) \$ git add templates/to_do.html git commit -m "Add Item Hash field to To-Do form" On branch master Untracked files: (use "git add ..." to include in what will be committed) todo.html.txt

nothing added to commit but untracked files present (use "git add" to track)

Mohds@Mohdsahal MINGW64 ~/flask-mongo-form (master) \$ git checkout -b main Switched to a new branch 'main'

Mohds@Mohdsahal MINGW64 ~/flask-mongo-form (main) \$ git merge master Already up to date.

Mohds@Mohdsahal MINGW64 ~/flask-mongo-form (main) \$ git log --oneline 86b7ba4 (HEAD -> main, master) Add Item ID field to To-Do form

Mohds@Mohdsahal MINGW64 ~/flask-mongo-form (main) \$ git checkout main git reset --soft 86b7ba4 Already on 'main'

Mohds@Mohdsahal MINGW64 ~/flask-mongo-form (main) \$ git commit -m "Recommit To-Do form with only Item ID field" On branch main Untracked files: (use "git add ..." to include in what will be committed) todo.html.txt

nothing added to commit but untracked files present (use "git add" to track)

Mohds@Mohdsahal MINGW64 ~/flask-mongo-form (main) \$ git checkout master Switched to branch 'master'

Mohds@Mohdsahal MINGW64 ~/flask-mongo-form (master) \$ git rebase main Current branch master is up to date.

Mohds@Mohdsahal MINGW64 ~/flask-mongo-form (master) \$ git add templates/to_do.html git rebase --continue fatal: no rebase in progress

Mohds@Mohdsahal MINGW64 ~/flask-mongo-form (master) \$ git log --oneline 86b7ba4 (HEAD -> master, main) Add Item ID field to To-Do form

Mohds@Mohdsahal MINGW64 ~/flask-mongo-form (master) \$ git add templates/to_do.html git commit -m "Add Item UUID field to To-Do form" On branch master Untracked files: (use "git add ..." to include in what will be committed) todo.html.txt

nothing added to commit but untracked files present (use "git add" to track)

Mohds@Mohdsahal MINGW64 ~/flask-mongo-form (master) \$ git add templates/to_do.html

Mohds@Mohdsahal MINGW64 ~/flask-mongo-form (master) \$ git commit -m "Add Item UUID field to To-Do form" On branch master Untracked files: (use "git add ..." to include in what will be committed) todo.html.txt

nothing added to commit but untracked files present (use "git add" to track)

Mohds@Mohdsahal MINGW64 ~/flask-mongo-form (master) \$ git add templates/to_do.html

Mohds@Mohdsahal MINGW64 ~/flask-mongo-form (master) \$ git commit -m "Add Item UUID field to To-Do form" [master d8b74f0] Add Item UUID field to To-Do form 1 file changed, 9 insertions(+), 3 deletions(-)

Mohds@Mohdsahal MINGW64 ~/flask-mongo-form (master) \$ git add templates/to_do.html git commit -m "Add Item Hash field to To-Do form" [master 9302e85] Add Item Hash field to To-Do form 1 file changed, 3 insertions(+)

Mohds@Mohdsahal MINGW64 ~/flask-mongo-form (master) \$ git checkout -b main fatal: a branch named 'main' already exists

Mohds@Mohdsahal MINGW64 ~/flask-mongo-form (master) \$ git checkout main Switched to branch 'main'

Mohds@Mohdsahal MINGW64 ~/flask-mongo-form (main) \$ git log --oneline 86b7ba4 (HEAD -> main) Add Item ID field to To-Do form

Mohds@Mohdsahal MINGW64 ~/flask-mongo-form (main) \$ git reset --hard 86b7ba4 HEAD is now at 86b7ba4 Add Item ID field to To-Do form

Mohds@Mohdsahal MINGW64 ~/flask-mongo-form (main) \$ git log --oneline 86b7ba4 (HEAD -> main) Add Item ID field to To-Do form

Mohds@Mohdsahal MINGW64 ~/flask-mongo-form (main) \$ git checkout master Switched to branch 'master'

Mohds@Mohdsahal MINGW64 ~/flask-mongo-form (master) \$ git rebase main Current branch master is up to date.

Mohds@Mohdsahal MINGW64 ~/flask-mongo-form (master) \$ git log --oneline 9302e85 (HEAD -> master) Add Item Hash field to To-Do form d8b74f0 Add Item UUID field to To-Do form 86b7ba4 (main) Add Item ID field to To-Do form

Mohds@Mohdsahal MINGW64 ~/flask-mongo-form (master) \$

Screenshots

```
app.py M X todo.html
app.py > ...
1 # Import required libraries
2 from flask import Flask, render_template, request, redirect, url_for
3 from pymongo import MongoClient
4 import certifi
5 import os
6
7 # Load MongoDB URI from environment variable
8 MONGO_URI = os.getenv(
9     "MONGO_URI",
10    "mongodb+srv://sahalusr1:securepass@cluster0.bdnym.mongodb.net/?retryWrites=true&w=majority&appName=Cluster0"
11 )
12
13 # Create the Flask app
14 app = Flask(__name__)
15
16 # Route to handle form submission
17 @app.route("/submittodoitem", methods=["POST"])
18 def submit_todo_item():
19     """
20     Handles form submission from your HTML form.
21     Inserts the submitted itemName and itemDescription into MongoDB.
22     """
23     # Get the submitted data from the form
24     item_name = request.form.get("itemName")
25     item_description = request.form.get("itemDescription")
26
27     if item_name and item_description:
28         # Connect to MongoDB
29         client = MongoClient(MONGO_URI, tlsCAFile=certifi.where())
30         db = client["todo_database"]
31         collection = db["todos"]
32
33         # Insert the submitted item into MongoDB
34         collection.insert_one({
35             "itemName": item_name,
36             "itemDescription": item_description
37         })
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS C:\Users\Mohds\flask-mongo-form> & C:\Users\Mohds\AppData\Local\Programs\Python\Python313\python.exe c:/Users/Mohds/flask-mongo-form/app.py
* Serving Flask app 'app'
* Debug mode: on
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
* Running on http://127.0.0.1:5000
Press CTRL+C to quit
* Restarting with stat
* Debugger is active!
* Debugger PIN: 375-489-153
127.0.0.1 - - [25/Aug/2025 16:02:39] "GET / HTTP/1.1" 200 -
127.0.0.1 - - [25/Aug/2025 16:02:53] "POST /submittodoitem HTTP/1.1" 302 -
127.0.0.1 - - [25/Aug/2025 16:02:54] "GET / HTTP/1.1" 200 -

```

Not Committed Yet Ln 19, Col 8 Spaces: 4 UTF-8 C

```

app.py > ...
33     # Insert the submitted item into MongoDB
34     collection.insert_one({
35         "name": item_name,
36         "description": item_description
37     })
38
39     # Redirect back to the home page showing all tasks
40     return redirect(url_for("show_todo_list"))
41
42 # Route to display the todo list
43 @app.route("/")
44 def show_todo_list():
45     # Connect to MongoDB
46     client = MongoClient(MONGO_URI, tlsCAFile=certifi.where())
47     db = client["todo_database"]
48     collection = db["todos"]
49
50     # Fetch all tasks
51     tasks = collection.find()
52
53     # Render tasks to HTML template
54     return render_template("todo.html", tasks=tasks)
55
56 # Run the Flask app
57 if __name__ == "__main__":
58     app.run(debug=True)
59
60 from flask import Flask, render_template, request, redirect, url_for
61 from pymongo import MongoClient
62
63 app = Flask(__name__)
64
65 # MongoDB connection
66 client = MongoClient("mongodb+srv://sahalusr1:securepass@cluster0.bdnym.mongodb.net/?retryWrites=true&w=majority&appName=Cluster0") # replace with your URI

```

PROBLEMS OUTPUT DEBUG CONSOLE **TERMINAL** PORTS

```

PS C:\Users\Mohds\flask-mongo-form> & C:/Users/Mohds/AppData/Local/Programs/Python/Python313/python.exe c:/Users/Mohds/flask-mongo-form/app.py
* Serving Flask app 'app'
* Debug mode: on
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
* Running on http://127.0.0.1:5000
Press CTRL+C to quit
* Restarting with stat
* Debugger is active!
* Debugger PIN: 375-489-153
127.0.0.1 - - [25/Aug/2025 16:02:39] "GET / HTTP/1.1" 200 -
127.0.0.1 - - [25/Aug/2025 16:02:53] "POST /submittodoitem HTTP/1.1" 302 -
127.0.0.1 - - [25/Aug/2025 16:02:54] "GET / HTTP/1.1" 200 -

```

```
app.py > ...
67 db = client['todo_db']
68 collection = db['tasks']
69
70 @app.route("/")
71 def index():
72     tasks = list(collection.find())
73     return render_template("todo.html", tasks=tasks)
74
75 @app.route("/submittodoitem", methods=["POST"])
76 def submit_todo_item():
77     # collect all fields
78     itemID = request.form.get("itemID")
79     itemUUID = request.form.get("itemUUID")
80     itemHash = request.form.get("itemHash")
81     name = request.form.get("itemName")
82     description = request.form.get("itemDescription")
83
84     # insert into MongoDB
85     collection.insert_one({
86         "itemID": itemID,
87         "itemUUID": itemUUID,
88         "itemHash": itemHash,
89         "name": name,
90         "description": description
91     })
92
93     return redirect(url_for("index"))
94
95 if __name__ == "__main__":
96     app.run(debug=True)
97
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS C:\Users\Mohds\flask-mongo-form> & C:/Users/Mohds/AppData/Local/Programs/Python/Python313/python.exe c:/Users/Mohds/flask-mongo-form/app.py

* Serving Flask app 'app'

* Debug mode: on

WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.

* Running on http://127.0.0.1:5000

Press CTRL+C to quit

* Restarting with stat

* Debugger is active!

* Debugger PIN: 375-489-153

127.0.0.1 - - [25/Aug/2025 16:02:39] "GET / HTTP/1.1" 200 -

127.0.0.1 - - [25/Aug/2025 16:02:53] "POST /submittodoitem HTTP/1.1" 302 -

127.0.0.1 - - [25/Aug/2025 16:02:54] "GET / HTTP/1.1" 200 -

█


```
app.py M todo.html
templates > todo.html > html > body.container.mt-5 > ul.list-group > li.list-group-item > div.mb-2
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1.0">
6   <title>Todo List</title>
7   <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.min.css" rel="stylesheet">
8 </head>
9 <body class="container mt-5">
10
11   <h1 class="mb-4 text-center">Todo List</h1>
12
13   <!-- Form to add new todo -->
14   <form action="{{ url_for('submit_todo_item') }}" method="post" class="mb-5">
15     <div class="row g-3">
16       <div class="col-md-4">
17         <input type="text" name="itemID" class="form-control" placeholder="Item ID" required>
18       </div>
19       <div class="col-md-4">
20         <input type="text" name="itemUUID" class="form-control" placeholder="Item UUID" required>
21       </div>
22       <div class="col-md-4">
23         <input type="text" name="itemHash" class="form-control" placeholder="Item Hash" required>
24       </div>
25     </div>
26     <div class="mt-3">
27       <input type="text" name="itemName" class="form-control mb-3" placeholder="Item Name" required>
28       <textarea name="itemDescription" class="form-control" placeholder="Item Description" rows="3" required></textarea>
29     </div>
30     <button type="submit" class="btn btn-primary mt-3">Submit</button>
31   </form>
32
33   <!-- Display todo list -->
34   <ul class="list-group">
```

```
app.py M todo.html
templates > todo.html > html > body.container.mt-5 > ul.list-group > li.list-group-item > div.mb-2
2 <html lang="en">
9 <body class="container mt-5">
14   <form action="{{ url_for('submit_todo_item') }}" method="post" class="mb-5">
15     <button type="submit" class="btn btn-primary mt-3">Submit</button>
31   </form>
32
33   <!-- Display todo list -->
34   <ul class="list-group">
35     {% for task in tasks %}
36       <li class="list-group-item">
37         <div class="mb-2">
38           <span class="badge bg-secondary">ID: {{ task.itemID }}</span>
39           <span class="badge bg-info text-dark">UUID: {{ task.itemUUID }}</span>
40           <span class="badge bg-warning text-dark">Hash: {{ task.itemHash }}</span>
41         </div>
42         <strong>{{ task.name }}</strong>: {{ task.description }}
43       </li>
44     {% endfor %}
45   </ul>
46
47 </body>
48 </html>
49

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS C:\Users\Mohds\flask-mongo-form> & C:\Users\Mohds\AppData\Local\Programs\Python\Python313\python.exe c:/Users/Mohds/flask-mongo-form/app.py
* Serving Flask app 'app'
* Debug mode: on
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
* Running on http://127.0.0.1:5000
Press CTRL+C to quit
* Restarting with stat
* Debugger is active!
* Debugger PIN: 375-489-153
127.0.0.1 - - [25/Aug/2025 16:02:39] "GET / HTTP/1.1" 200 -
127.0.0.1 - - [25/Aug/2025 16:02:53] "POST /submittodoitem HTTP/1.1" 302 -
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS C:\Users\Mohds\flask-mongo-form> & C:/Users/Mohds/AppData/Local/Programs/Python/Python313/python.exe c:/Users/Mohds/flask-mongo-form/app.py
* Serving Flask app 'app'
* Debug mode: on
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
* Running on http://127.0.0.1:5000
Press CTRL+C to quit
* Restarting with stat
* Debugger is active!
* Debugger PIN: 375-489-153
```

Mohds@Mohdsahal MINGW64 ~/flask-mongo-form (master_1)

```
● $ git checkout master_1
M      app.py
Already on 'master_1'
```

Mohds@Mohdsahal MINGW64 ~/flask-mongo-form (master_1)

```
⊗ $ git add templates/todo.html
git commit -m "Added Item ID field"
On branch master_1
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   app.py

no changes added to commit (use "git add" and/or "git commit -a")
```

Mohds@Mohdsahal MINGW64 ~/flask-mongo-form (master_1)

```
$ git add templates/todo.html
⊗ git commit -m "Added Item UUID field"
On branch master_1
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   app.py

no changes added to commit (use "git add" and/or "git commit -a")
```

Mohds@Mohdsahal MINGW64 ~/flask-mongo-form (master_1)

```
$ git add templates/todo.html
git commit -m "Added Item Hash field"
On branch master_1
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   app.py

no changes added to commit (use "git add" and/or "git commit -a")
```

```
Mohds@Mohdsahal MINGW64 ~/flask-mongo-form (master_1)
```

```
$ git push origin master_1
```

```
Everything up-to-date
```

```
Mohds@Mohdsahal MINGW64 ~/flask-mongo-form (master_1)
```

```
$ git checkout main
```

```
git merge master_1
```

```
M      app.py
```

```
Switched to branch 'main'
```

```
Your branch is up to date with 'origin/main'.
```

```
Already up to date.
```

```
Mohds@Mohdsahal MINGW64 ~/flask-mongo-form (main)
```

```
$ git log # find commit hash for "Added Item ID field"
```

```
git reset --soft <commit_hash>
```

```
git commit -m "Revert to only Item ID field"
```

```
git push origin main --force
```

```
commit 731565f2426624b391492d1ba80772f5d53a1c1c (HEAD -> main, origin/master_1, origin/main, master_1)
```

```
Author: Sahal <your@email.com>
```

```
Date: Mon Aug 25 15:50:57 2025 +0530
```

```
Added Item ID field
```

```
commit a06eafb010a931d01e56d258f568611daee06887 (origin/sahal, sahal_new, sahal)
```

```
Author: Sahal <your@email.com>
```

```
Date: Mon Aug 25 12:32:48 2025 +0530
```

```
Initial commit with Flask project files
```

```
bash: syntax error near unexpected token `newline'
```

```
On branch main
```

```
Your branch is up to date with 'origin/main'.
```

```
Changes not staged for commit:
```

```
(use "git add <file>..." to update what will be committed)
```

```
(use "git restore <file>..." to discard changes in working directory)
```

```
modified:   app.py
```

```
no changes added to commit (use "git add" and/or "git commit -a")
```

```
Everything up-to-date
```

```
Mohds@Mohdsahal MINGW64 ~/flask-mongo-form (main)
```

```
$ git checkout master_1
```

```
git rebase main
```

```
git push origin master_1 --force
```

```
M      app.py
```

```
Switched to branch 'master_1'
```

```
error: cannot rebase: You have unstaged changes.
```

```
error: Please commit or stash them.
```

```
Everything up-to-date
```

Todo List

Item ID
Item UUID
Item UUID
Item Hash
Item Hash
Item Name
Item Description
Submit
zxcz: zxczdz
orange: dfgdfg
apple: apple is red
apple: dssdf
apple: apple is red
apple: apple is red
sadfsf: dgfsdfsdf
sadfsf: red is red
orange: orange is orange

```
1 <!DOCTYPE html>
2 <html lang="en">
3
4 <head>
5   <meta charset="UTF-8">
6   <title>Todo List</title>
7   <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.min.css" rel="stylesheet">
8 </head>
9
10 <body class="container mt-5">
11
12   <h1 class="mb-4">Todo List</h1>
13
14   <!-- Form to add new todo -->
15   <form action="{{ url_for('submit_todo_item') }}" method="post" class="mb-4">
16
17     <div class="mb-3">
18       <input type="text" name="itemID" class="form-control" placeholder="Item ID" required>
19     </div>
20
21     <div class="mb-3">
22       <input type="text" name="itemUUID" class="form-control" placeholder="Item UUID" required>
23     </div>
24
25     <div class="mb-3">
26       <input type="text" name="itemUUID" class="form-control" placeholder="Item UUID" required>
27     </div>
28
29     <div class="mb-3">|
30       <input type="text" name="itemHash" class="form-control" placeholder="Item Hash" required>
31     </div>
32
33
34     <div class="mb-3">
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
Press CTRL+C to quit
* Restarting with stat
* Debugger is active!
* Debugger PIN: 375-489-153
127.0.0.1 - - [26/Aug/2025 09:23:50] "GET / HTTP/1.1" 200 -
127.0.0.1 - - [26/Aug/2025 09:23:51] "GET /favicon.ico HTTP/1.1" 404 -
127.0.0.1 - - [26/Aug/2025 09:24:11] "POST /submittodoitem HTTP/1.1" 302 -
127.0.0.1 - - [26/Aug/2025 09:24:11] "GET / HTTP/1.1" 200 -
127.0.0.1 - - [26/Aug/2025 09:24:46] "POST /submittodoitem HTTP/1.1" 302 -
127.0.0.1 - - [26/Aug/2025 09:24:46] "GET / HTTP/1.1" 200 -
127.0.0.1 - - [26/Aug/2025 09:25:08] "POST /submittodoitem HTTP/1.1" 302 -
127.0.0.1 - - [26/Aug/2025 09:25:09] "GET / HTTP/1.1" 200 -
```

```
33
34     <div class="mb-3">
35         <input type="text" name="itemHash" class="form-control" placeholder="Item Hash" required>
36     </div>
37
38     <div class="mb-3">
39         <input type="text" name="itemName" class="form-control" placeholder="Item Name" required>
40     </div>
41
42     <div class="mb-3">
43         <textarea name="itemDescription" class="form-control" placeholder="Item Description" required></textarea>
44     </div>
45
46     <button type="submit" class="btn btn-primary">Submit</button>
47 </form>
48
49 <!-- Display todo list -->
50 <ul class="list-group">
51     {% for task in tasks %}
52     <li class="list-group-item">
53         <strong>{{ task.name }}</strong>: {{ task.description }}
54     </li>
55     {% endfor %}
56 </ul>
57
58 </body>
59
60 </html>
61
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

Press CTRL+C to quit
* Restarting with stat
* Debugger is active!
* Debugger PIN: 375-489-153

```
127.0.0.1 - - [26/Aug/2025 09:23:50] "GET / HTTP/1.1" 200 -
127.0.0.1 - - [26/Aug/2025 09:23:51] "GET /favicon.ico HTTP/1.1" 404 -
127.0.0.1 - - [26/Aug/2025 09:24:11] "POST /submittodoitem HTTP/1.1" 302 -
127.0.0.1 - - [26/Aug/2025 09:24:11] "GET / HTTP/1.1" 200 -
127.0.0.1 - - [26/Aug/2025 09:24:46] "POST /submittodoitem HTTP/1.1" 302 -
127.0.0.1 - - [26/Aug/2025 09:24:46] "GET / HTTP/1.1" 200 -
127.0.0.1 - - [26/Aug/2025 09:25:08] "POST /submittodoitem HTTP/1.1" 302 -
127.0.0.1 - - [26/Aug/2025 09:25:09] "GET / HTTP/1.1" 200 -
```

```
<div>
</div>
to_do.html M app.py X
app.py > ...
1 from flask import Flask, render_template, request, redirect, url_for
2 from pymongo import MongoClient
3 import certifi
4 import os
5
6 # Load MongoDB URI from environment variable or use default
7 MONGO_URI = os.getenv(
8     "MONGO_URI",
9     "mongodb+srv://sahalus1:securepass@cluster0.bdnymr.mongodb.net/?retryWrites=true&w=majority&appName=Cluster0"
10 )
11
12 # Initialize Flask app
13 app = Flask(__name__)
14
15 # Initialize MongoDB client and collection
16 client = MongoClient(MONGO_URI, tlsCAFile=certifi.where())
17 db = client["todo database"]
18 collection = db["todos"]
19
20 # Route to display the todo list
21 @app.route("/")
22 def index():
23     tasks = list(collection.find())
24     return render_template("to_do.html", tasks=tasks)
25
26 # Route to handle form submission
27 @app.route("/submittodoitem", methods=["POST"])
28 def submit_todo_item():
29     # Collect all fields from form
30     itemID = request.form.get("itemID")
31     itemUUID = request.form.get("itemUUID")
32     itemHash = request.form.get("itemHash")
33     itemName = request.form.get("itemName")
34     itemDescription = request.form.get("itemDescription")
35
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

Press CTRL+C to quit
* Restarting with stat
* Debugger is active!
* Debugger PIN: 375-489-153

```
127.0.0.1 - - [26/Aug/2025 09:23:50] "GET / HTTP/1.1" 200 -
127.0.0.1 - - [26/Aug/2025 09:23:51] "GET /favicon.ico HTTP/1.1" 404 -
127.0.0.1 - - [26/Aug/2025 09:24:11] "POST /submittodoitem HTTP/1.1" 302 -
127.0.0.1 - - [26/Aug/2025 09:24:11] "GET / HTTP/1.1" 200 -
127.0.0.1 - - [26/Aug/2025 09:24:46] "POST /submittodoitem HTTP/1.1" 302 -
127.0.0.1 - - [26/Aug/2025 09:24:46] "GET / HTTP/1.1" 200 -
127.0.0.1 - - [26/Aug/2025 09:25:08] "POST /submittodoitem HTTP/1.1" 302 -
127.0.0.1 - - [26/Aug/2025 09:25:09] "GET / HTTP/1.1" 200 -
```

```
to_do.html M  app.py X
app.py > ...
28 def submit_todo_item():
34     itemDescription = request.form.get("itemDescription")
35
36     if itemID and itemUUID and itemHash and itemName and itemDescription:
37         # Insert into MongoDB
38         collection.insert_one({
39             "itemID": itemID,
40             "itemUUID": itemUUID,
41             "itemHash": itemHash,
42             "name": itemName,
43             "description": itemDescription
44         })
45
46     # Redirect back to the todo list page
47     return redirect(url_for("index"))
48
49 # Run the Flask app
50 if __name__ == "__main__":
51     app.run(debug=True)
52
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

Press CTRL+C to quit
* Restarting with stat
* Debugger is active!
* Debugger PIN: 375-489-153

```
127.0.0.1 - - [26/Aug/2025 09:23:50] "GET / HTTP/1.1" 200 -
127.0.0.1 - - [26/Aug/2025 09:23:51] "GET /favicon.ico HTTP/1.1" 404 -
127.0.0.1 - - [26/Aug/2025 09:24:11] "POST /submittodoitem HTTP/1.1" 302 -
127.0.0.1 - - [26/Aug/2025 09:24:11] "GET / HTTP/1.1" 200 -
127.0.0.1 - - [26/Aug/2025 09:24:46] "POST /submittodoitem HTTP/1.1" 302 -
127.0.0.1 - - [26/Aug/2025 09:24:46] "GET / HTTP/1.1" 200 -
127.0.0.1 - - [26/Aug/2025 09:25:08] "POST /submittodoitem HTTP/1.1" 302 -
127.0.0.1 - - [26/Aug/2025 09:25:09] "GET / HTTP/1.1" 200 -
```

```
Mohds@Mohdsahal MINGW64 ~/flask-mongo-form (master)
• $ git rebase main
Current branch master is up to date.

Mohds@Mohdsahal MINGW64 ~/flask-mongo-form (master)
• $ git log --oneline
9302e85 (HEAD -> master) Add Item Hash field to To-Do form
d8b74f0 Add Item UUID field to To-Do form
86b7ba4 (main) Add Item ID field to To-Do form

Mohds@Mohdsahal MINGW64 ~/flask-mongo-form (master)
○ $
```

```
Mohds@Mohdsahal MINGW64 ~/flask-mongo-form (main)
• $ git reset --hard 86b7ba4
HEAD is now at 86b7ba4 Add Item ID field to To-Do form

Mohds@Mohdsahal MINGW64 ~/flask-mongo-form (main)
• $ git log --oneline
86b7ba4 (HEAD -> main) Add Item ID field to To-Do form

Mohds@Mohdsahal MINGW64 ~/flask-mongo-form (main)
• $ git checkout master
Switched to branch 'master'
```

```

Mohds@Mohdsahal MINGW64 ~/flask-mongo-form (master)
❌ $ git checkout -b main
fatal: a branch named 'main' already exists

Mohds@Mohdsahal MINGW64 ~/flask-mongo-form (master)
● $ git checkout main
Switched to branch 'main'

Mohds@Mohdsahal MINGW64 ~/flask-mongo-form (main)
● $ git log --oneline
86b7ba4 (HEAD -> main) Add Item ID field to To-Do form

```

```

Mohds@Mohdsahal MINGW64 ~/flask-mongo-form (master)
$ git add templates/to_do.html
● git commit -m "Add Item Hash field to To-Do form"
[master 9302e85] Add Item Hash field to To-Do form
1 file changed, 3 insertions(+)

Mohds@Mohdsahal MINGW64 ~/flask-mongo-form (master)
❌ $ git checkout -b main
fatal: a branch named 'main' already exists

Mohds@Mohdsahal MINGW64 ~/flask-mongo-form (master)
● $ git checkout main
Switched to branch 'main'

```

```

Mohds@Mohdsahal MINGW64 ~/flask-mongo-form (master)
● $ git rebase main
Current branch master is up to date.

Mohds@Mohdsahal MINGW64 ~/flask-mongo-form (master)
● $ git log --oneline
9302e85 (HEAD -> master) Add Item Hash field to To-Do form
d8b74f0 Add Item UUID field to To-Do form
86b7ba4 (main) Add Item ID field to To-Do form

Mohds@Mohdsahal MINGW64 ~/flask-mongo-form (master)
○ $

```

```

$ git log --oneline
86b7ba4 (HEAD -> main) Add Item ID field to To-Do form

Mohds@Mohdsahal MINGW64 ~/flask-mongo-form (main)
$ git checkout master
Switched to branch 'master'

Mohds@Mohdsahal MINGW64 ~/flask-mongo-form (master)
$ git rebase main
Current branch master is up to date.

Mohds@Mohdsahal MINGW64 ~/flask-mongo-form (master)
$ git log --oneline

```