Guide: Pushing a Flask Application to GitHub

1. Install Git

Download and install Git from the official site:
🔗 https://git-scm.com/downloads

Verify the installation by running the following command in the command prompt:

```
git --version
```

2. Configure Git (First-Time Setup Only)

Before using Git, configure your user information so that it appears in commit records:

```
git config --global user.name "Your Name"
git config --global user.email "your-email@example.com"
```

Note:

- The `--global` flag applies this configuration to all repositories.
- If omitted, the settings will be applied only to the current repository.

To Check Your Git Configuration:

- **List all configurations:**
  - `git config --list`
- **Display configured username:**
  - `git config user.name`
- **Display configured email:**
  - `git config user.email`

3. Create a GitHub Account

Sign up on GitHub if you don't have an account.

4. Create a New Repository on GitHub

- **Click on New Repository**
- **Set the repository name as:**

  <BatchName>_<ProjectTitle>_Repo

Example:

G32_OnlineQuizApp_Repo

5. Clone the Repository to Your Local System

- **Copy the repository HTTPS or SSH URL from GitHub and run:**

  git clone <repository-url>

- This will create a local copy of the repository.

📝 Note: The cloned repository will contain only the files that are already in the GitHub repository.

6. Set Up a `.gitignore` File

- Inside the main project folder (in VS Code or any editor), create a file named `.gitignore` and add the following lines:

```
# Exclude virtual environment folder
env/

# Exclude database file
```

app.db

7. Freeze Project Dependencies

- **Generate a `requirements.txt` file to list all dependencies:**

  pip freeze > requirements.txt

8. Move Your Project Into the Cloned Repository Folder:

- Paste your Flask project folder inside the cloned repository folder.
- Ensure your main application files, `.gitignore`, and `requirements.txt` are inside the cloned repository.

9. Push the Flask Project to GitHub:

- Open the command prompt, navigate to the cloned repository, and run the following commands:

a. Add all project files to the staging area:

   git add .

b. Commit the staged files to the local repository:

   git commit -m "Initial project upload"

c. c. Push the changes to GitHub:

   git push origin main

(If your repository uses `master` instead of `main`, replace `main` with `master`.)