Lab Notebook

Team 9

University Details

Maulana Abul Kalam Azad University of Technology

Assignment Details

- **Assignment:** Create a Git Repository Containing Lab Notebook in a LaTeX File
- **Subject:** Software Tools and Techniques
- **Team no.:** 9
- **GitHub Repo Link:** `https://github.com/Dip9143/Lab-Notebook-Group-9`

Team Members

- **Member 1 (Lead):**
  - Name: Dip Kumar Majumder
  - Reg no.: 233002410605
  - Course: BSc in IT (Data Science)
  - GitHub link: `https://github.com/Dip9143`
- **Member 2:**
  - Name: Sagnik Halder
  - Reg no.: 233002410554
  - Course: BSc in IT (Artificial Intelligence)
  - GitHub link: `https://github.com/InfiniteParadoxicalExistence`
- **Member 3:**
  - Name: Barbie Doley
  - Reg no.: 233002410034
  - Course: BSc in Forensic Science
  - GitHub link: `https://github.com/BarbieDoley`
- **Member 4:**

  - Name: Azlan Jamshed
  - Reg no.: 233002410564
  - Course: BSc in IT (Artificial Intelligence)
  - GitHub link: `https://github.com/azlanjamshed`

- **Member 5:**
  - Name: Arindam Halder
  - Reg no.: 233002410573
  - Course: BSc in IT (Artificial Intelligence)
  - GitHub link: `https://github.com/Arindam8698`

Acknowledgment

We would like to express our sincere gratitude to our esteemed professors, **Mr. Ayan Ghosh** and **Mr. Pabitra Pal**, for their invaluable guidance, encouragement, and support throughout the course of this project. Their insights and feedback were instrumental in completing this work, and we are deeply grateful for their mentorship.

We also extend our heartfelt thanks to the Maulana Abul Kalam Azad University of Technology for providing us with the opportunity to work on this project, allowing us to enhance our skills and knowledge in the field of software tools and techniques.

Candidate's Signature	Registration Number

Contents

1 Lab Assignment 1: Calculator Program

Task: Create a local repository, build a C program for a calculator in the local repository, commit the changes, and publish it as a public repository on GitHub.

Procedure

1. **Initialize Local Repository:**
   - Open the terminal (or command prompt) and navigate to the directory where you want to create your project.
   - Run the command: `git init` to initialize a new Git repository.
2. **Create the C Program:**
   - Create a new file named `calculator.c` in your project directory.
   - Write the C code for the calculator program, ensuring it can perform basic arithmetic operations like addition, subtraction, multiplication, and division.
   - Save the file.
3. **Stage and Commit Changes:**
   - Stage the file for commit by running: `git add calculator.c`
   - Commit the file with a descriptive message: `git commit -m "Add basic calculator program"`
4. **Publish on GitHub:**
   - Log in to your GitHub account and create a new public repository.
   - In the terminal, link your local repository to the remote GitHub repository by running: `git remote add origin <repository-URL>`
   - Push your local commits to GitHub with the command: `git push -u origin main`
5. **Verify the Repository:**
   - Open your GitHub repository in a web browser to ensure the `calculator.c` file is present and the commit message is correctly displayed.

2 Lab Assignment 2: Mind Reader Application

Task: Your professor created a mind reader application and wants you to try it out. After running the program, you found the submit button looks dull. You renamed it "Chin Tapak Dum Dum," but the button became disproportionate. Your task is to fix the button issue and create a pull request with the solution.

Procedure

1. **Clone the Repository:**
    - Open GitHub Desktop or use the terminal to clone the repository: `https://github.com/GeekAyan/STT`
    - Run the command: `git clone https://github.com/GeekAyan/STT.git`
    - Navigate to the project directory.
2. **Run the Application:**
    - Follow the instructions provided in the `README.md` file to set up and run the mind reader application using your preferred Integrated Development Environment (IDE).
    - Observe the application's user interface, particularly the submit button.
3. **Identify and Rename the Button:**
    - Locate the submit button code in the application's source files.
    - Rename the button text to "Chin Tapak Dum Dum."
    - Notice that the button has become disproportionate due to the increased text length.
4. **Fix the Button Size:**
    - Analyze the layout code that controls the button's appearance.
    - Adjust the width and height properties, or use appropriate CSS/JavaFX adjustments to make the button proportionate.
    - Test the application to ensure the button now displays correctly and does not affect other UI elements.
5. **Commit and Push the Changes:**
    - Stage the modified files with: `git add .`
    - Commit the changes with a descriptive message: `git commit -m "Fix button size after renaming to 'Chin Tapak Dum Dum'"`
    - Push the changes to your forked repository on GitHub.
6. **Create a Pull Request:**
    - Go to your GitHub repository and click on "Compare & pull request."
    - Write a brief description of the changes made and submit the pull request to the original repository.
7. **Review and Merge:**
    - Wait for the repository owner to review your pull request.
    - If accepted, your changes will be merged into the main project.

3 Lab Assignment 3: Git Branching, Merging, and Conflict Resolution

Task: Demonstrate proficiency in Git branching, merging, and conflict resolution in a step-by-step process.

Procedure

1. **Create a GitHub Repository:**
   - Create a new repository called `git-advanced` on GitHub.
2. **Clone the Repository:**
   - Clone the repository to your local machine using the command: `git clone <repository-url>`
3. **Create and Switch to a New Branch (feature-1):**
   - Use the command `git checkout -b feature-1` to create and switch to a new branch named `feature-1`.
4. **Add and Commit Changes on feature-1:**
   - Create a file `shared.txt` and add the content:

     ```
     This is a shared file.
     Line 1: Original text.
     Line 2: Original text.
     ```

   - Stage and commit the changes: `git commit -m "Add shared.txt with original text"`
5. **Push the Branch to GitHub:**
   - Push the `feature-1` branch to GitHub: `git push origin feature-1`
6. **Create Another Branch (feature-2):**
   - Switch to `feature-2` branch using the command: `git checkout -b feature-2`
7. **Modify the Shared File on feature-2:**
   - Modify the second line of `shared.txt`:

     ```
     Line 2: Modified text in feature-2.
     ```

   - Stage and commit the changes: `git add shared.txt`
   - Commit with a descriptive message: `git commit -m "Modify Line 2 in feature-2"`
   - Push the changes: `git push origin feature-2`
8. **Switch Back to feature-1 and Modify:**

- Switch back to `feature-1` using: `git checkout feature-1`
- Modify the second line on `shared.txt`:

  ```
  Line 2: Modified text in feature-1.
  ```

- Stage and commit the changes: `git add shared.txt`
- Commit with a descriptive message: `git commit -m "Modify Line 2 in feature-1"`
- Push the changes: `git push origin feature-1`

9. **Merge feature-1 into main:**
   - Switch to the `main` branch: `git checkout main`
   - Merge the `feature-1` branch into the `main` branch: `git merge feature-1`
   - Push the merged changes: `git push origin main`
10. **Merge feature-2 and Handle Conflict:**
    - Merge `feature-2` into `main`: `git merge feature-2`
    - Git will notify you of a merge conflict in `shared.txt`.
    - Open `shared.txt` in your text editor and resolve the conflict by choosing the appropriate changes or combining them.
    - After resolving, stage the resolved file: `git add shared.txt`
    - Commit the merge: `git commit -m "Merge feature-2 into main and resolve conflicts"`
    - Push the resolved main branch to GitHub: `git push origin main`
11. **Clean Up Branches:**
    - Delete both `feature-1` and `feature-2` branches locally:

      ```
      git branch -d feature-1
      git branch -d feature-2
      ```

    - Delete the branches on GitHub:

      ```
      git push origin --delete feature-1
      git push origin --delete feature-2
      ```

4 Lab Assignment 4: Create a CV Using LaTeX

Task: Create a CV using a LaTeX document.

Procedure

1. **Outline Your CV Content:**
    - Include your name, contact details, and a professional summary.
    - List academic qualifications, work experience, skills, projects, and certifications.
2. **Decide on the Structure and Layout:**
    - Organize the CV into sections such as Personal Information, Experience, Education, etc.
3. **Choose a LaTeX Template:**
    - Select a template that suits your style from Overleaf or a LaTeX library.
4. **Customize the Template:**
    - Edit the template with your personal content (experience, qualifications, etc.).
5. **Adjust Formatting:**
    - Ensure consistency in fonts and section headings.
6. **Proofread and Finalize:**
    - Review for any errors or formatting issues.
    - Ensure alignment and organization of sections.
7. **Compile and Export:**
    - Compile the LaTeX document and export it as a PDF for sharing.

5 Lab Assignment 5: Solution for Creating a ZIP File with LaTeX Source, PDF, and Image

Task: Create a ZIP file containing LaTeX source code (.tex), the generated output (.pdf), and an image file (.png or .jpg).

Procedure

1. **Create the LaTeX Source Code (.tex):**
    - Open a LaTeX editor (like Overleaf or TeXShop).
    - Write the LaTeX code following the structure from the assignment, replicating the provided formatting from the PDF.
    - Replace placeholders such as "Your Name Here" and "Your Photo Here" with your actual information.

- Name the file according to the assignment's format. For example, if your roll number is 30084323006, department is IS, and your first name is Dip, name it: `30084323006_IS_Dip.tex`.

2. **Compile the LaTeX File to PDF:**
   - Once the LaTeX code is ready, compile the .tex file to generate the output document in .pdf format.
   - Download the generated PDF file.
3. **Include an Image File:**
   - Choose or create an appropriate image (either .png or .jpg) to include.
   - Ensure the image relates to your content and follows the assignment's expectations.
4. **Organize Files:**
   - You should now have three files:
     - `30084323006_IS_Dip.tex` (your LaTeX source code).
     - `30084323006_IS_Dip.pdf` (the compiled output of your LaTeX document).
     - Your image file (either .png or .jpg).
5. **Create the ZIP File:**
   - Collect the three files (LaTeX source, PDF output, and image).
   - Create a compressed ZIP archive and name it `30084323006_IS_Dip.zip`.
   - On Windows: Right-click on the files → "Send to" → "Compressed (zipped) folder".
   - On macOS: Select the files → Right-click → "Compress".
   - On Linux: Use the command `zip` in the terminal:

     ```
     zip 30084323006_IS_Dip.zip 30084323006_IS_Dip.tex 30084323006_IS_Dip.pdf imag
     ```

6. **Verify:**
   - Open the zip file and ensure it contains all three files: .tex, .pdf, and the image.
   - Ensure the zip file is named correctly.

6 Conclusion

Throughout the lab assignments, we gained hands-on experience in several crucial aspects of software development and collaboration using Git and LaTeX. By working on tasks such as building a calculator program, modifying an existing application, handling Git branching and conflict resolution, and creating a CV using LaTeX, we strengthened our technical proficiency in C programming, version control, and document preparation.

The process of resolving merge conflicts, in particular, provided valuable insights into collaborative development and how proper Git workflows can prevent and address issues

that arise when multiple team members work on the same project. Moreover, the task of customizing a LaTeX CV highlighted the advantages of using LaTeX for professional document preparation, given its flexibility and control over formatting.

These experiences not only enhanced our understanding of the technical concepts but also fostered teamwork, problem-solving, and critical thinking, which will be beneficial for future projects and professional work environments. We look forward to applying these skills to more complex and collaborative projects in the future.