MAULANA ABUL KALAM AZAD UNIVERSITY OF TECHNOLOGY

Software Tools and Technology (Lab Notbook)

Group 12		
NAME	ROLL NO.	DEPARTMENT
Sk Shoaib Akhter [LEAD]	30001223043	BCA
Urjjaswi Paul	30059223024	Bsc in Forensic Science
Pranjal Paul	30059223023	Bsc in Forensic Science
Rupkatha Bhowmik	30059223009	Bsc in Forensic Science
Alankrita Ghosh	30059223010	Bsc in Forensic Science

Table of Contents

Lab -1 by Shoaib

Introduction to GitHub and GitHub installation

GitHub

GitHub is a cloud-based platform specifically designed for developers to collaborate on code and manage software projects using Git, a widely used version control system. At its core, GitHub provides a space called repositories where code is stored, allowing users to track changes over time and maintain a complete history of all updates made to the project. Through GitHub, developers can work on different features or versions of a project simultaneously by creating separate branches, which can later be merged back into the main codebase using pull requests, ensuring smooth integration of changes.

Installation

Installing GitHub involves a few simple steps to get started with managing and collaborating on code. First, download and install Git, the version control system that GitHub uses, from the official Git website. For Windows users, run the downloaded installer and follow the prompts. On macOS, you can install Git through the terminal if it's not already present. For Linux users, Git can be installed via the terminal with package management commands. Once Git is set up, configure it with your GitHub username and email to link your commits to your GitHub account. If you prefer a graphical interface, you can also download GitHub Desktop from its official website, which makes it easier to handle repositories, make commits, and manage branches without using the command line. After installation, sign in with your GitHub credentials or create a new account if necessary.

Lab-2 by Pranjal

Enumerate ABC Format and Roman Number in LaTeX

Introduction

To create enumerated lists in LaTeX using both alphabetic (ABC) format and Roman numeral format, you can use the `enumerate` environment. LaTeX allows customization of list labels either by using the `enumerate` package (basic customization) or the `enumitem` package (more advanced customization). Below is a detailed explanation and corresponding LaTeX code.

1.Alphabetic Format (ABC):

- For alphabetic enumeration (A, B, C...), you can use either the `enumerate` package or the `enumitem` package.

2. Roman Numerals:

- For Roman numeral enumeration (I, II, III...), you can use `[I.]` with the `enumerate` package, or `label=.` with the `enumitem` package.

Example

Here's the LaTeX code that implements both alphabetic and Roman numeral enumerations:

```
\documentclass{article}

% Using the enumerate package for basic list formatting
\usepackage{enumerate}

% Using the enumitem package for advanced list formatting
\usepackage{enumitem}

\begin{document}

\section*{Enumerated Lists: ABC Format and Roman Numerals}

\subsection*{Using the enumerate Package}

\subsubsection*{Alphabetic Format (ABC)}

% Alphabetic enumeration using enumerate package
\begin{enumerate}[A.]
  \item First item
  \item Second item
```

```
  \item Third item
\end{enumerate}

\subsubsection*{Roman Numerals}

% Roman numeral enumeration using enumerate package
\begin{enumerate}[I.]
  \item First item
  \item Second item
  \item Third item
\end{enumerate}

\subsection*{Using the enumitem Package}

\subsubsection*{Alphabetic Format (ABC)}

% Alphabetic enumeration using enumitem package
\begin{enumerate}[label=\Alph*.]
  \item First item
  \item Second item
  \item Third item
\end{enumerate}

\subsubsection*{Roman Numerals}

% Roman numeral enumeration using enumitem package
\begin{enumerate}[label=\Roman*.]
  \item First item
  \item Second item
  \item Third item
\end{enumerate}

\end{document}
```

Explanation

1. Alphabetic Format (ABC):

- **Using the enumerate package:** In the `enumerate` environment, you can specify the list type as `[A.]` to generate an alphabetic list (A, B, C...).
- **Using the enumitem package:** The `enumitem` package allows more flexibility by setting the label format as `label=.`.

2. **Roman Numerals:**

- **Using the enumerate package:** For Roman numerals, you use `[I.]` to create the list items as I, II, III...
- **Using the enumitem package:** You can customize the label with `label=.` to achieve the same effect with more control over spacing and style.

Output:

The compiled LaTeX document will display two types of lists (ABC and Roman numerals) using both `enumerate` and `enumitem` packages.

Lab-3 by Rupkatha

1 Calculator Program using C

1.1 Objective

The objective of this lab is to develop a basic calculator program using the C programming language. The calculator will perform simple arithmetic operations like addition, subtraction, multiplication, and division based on user input.

1.2 Program Overview

The calculator program is designed to:

- Accept two numbers from the user.
- Prompt the user to select an arithmetic operation (Addition, Subtraction, Multiplication, Division).
- Perform the selected operation.
- Display the result of the operation to the user.

The program includes error handling to manage division by zero and other invalid inputs.

1.3 Code Implementation

```
#include<stdio.h>
#include<conio.h>
void main(){
    float a,b,c;
    char ch;
        printf("Enter the first number : ");
        scanf("%f",&a);
```

```
            printf("Enter user choice operation : ");
            scanf(" %c",&ch);
        printf("Enter the second number : ");
        scanf("%f",&b);
    switch(ch){
        case'+':c=a+b;
            printf("Output is : %f",c);
        break;
        case'-':c=a-b;
            printf("Output is : %f",c);
        break;
        case'*':c=a*b;
            printf("Output is : %f",c);
        break;
        case'/':c=a/b;
            printf("Output is : %f",c);
        break;
            default:printf("Invalid operation");
        break;
    } getch();
}
```

1.4 Compiling and Running the Program

To compile and run the calculator program:

1. Open a terminal or command prompt.
2. Navigate to the directory where the C file is located.
3. Compile the program using a C compiler (e.g., GCC):

```
gcc calculator.c -o calculator
```

4. Run the compiled program:

```
./calculator
```

1.5 Adding the Calculator Program to GitHub Repository

To add this calculator program to a GitHub repository, follow these steps:

1.5.1 Step 1: Initialize a Local Git Repository

1. Open the terminal and navigate to the directory where your `calculator.c` file is located.
2. If you haven't already, initialize a Git repository in that directory:

```
git init
```

   This command creates a new Git repository in the current directory.

1.5.2 Step 2: Add the File to the Repository

1. Add the `calculator.c` file to the staging area:

```
git add calculator.c
```

   This command stages the file, indicating that you want to include it in the next commit.

1.5.3 Step 3: Commit the Changes

1. Commit the file to the repository with a meaningful message:

```
git commit -m "Add calculator program in C"
```

1.5.4 Step 4: Push the Changes to GitHub

1. Link your local repository to a remote GitHub repository:

```
git remote add origin https://github.com/yourusername/your-repo-name.git
```

2. Push the changes to the GitHub repository:

```
git push -u origin master
```

1.5.5 Step 5: Verify the Upload

1. Go to your GitHub repository URL in a web browser.
2. Verify that the `calculator.c` file is listed and accessible in the repository.

Lab-4 by Urjjaswi

How to create Matrix in LaTeX

To create a matrix in LaTeX, we use the `amsmath` package, which provides various environments for displaying matrices. Here's a basic guide on how to create a matrix and the different options available:

- **Include the `amsmath` Package**

First, ensure you have the `amsmath` package included in your LaTeX document preamble. Add the following line:

```
\usepackage{amsmath}
```

- **Matrix Environments**

There are several environments for creating matrices, depending on how you want them formatted. Here are the most common ones:

- `matrix`: A basic matrix without brackets.
- `bmatrix`: A matrix with square brackets.
- `pmatrix`: A matrix with parentheses.
- `vmatrix`: A matrix with vertical bars.
- `Vmatrix`: A matrix with double vertical bars.

Syntax

The syntax for creating a matrix is similar across different environments. Use the `\begin{environment}` and `\end{environment}` commands to enclose the matrix content. Separate the elements in each row with `&` and end each row with

.

Here's an example of a 2x2 matrix in each environment:

```
[
$ \begin{matrix}
    1 & 0 \\
    0 & 1
\end{matrix}
$ ]

\begin{bmatrix}   [
 $   \begin{matrix}
    1 & 0\\
    0 & 1
    \end{matrix}
 $   ]

a_{11} & a_{12} \\
a_{21} & a_{22}
\end{bmatrix}

\begin{pmatrix}
a_{11} & a_{12} \\
a_{21} & a_{22}
```

```
\end{pmatrix}

\begin{vmatrix}
a_{11} & a_{12} \\
a_{21} & a_{22}
\end{vmatrix}

\begin{Vmatrix}
a_{11} & a_{12} \\
a_{21} & a_{22}
\end{Vmatrix}
```

Explanation

- `\begin{matrix} ... \end{matrix}`: Creates a matrix with no surrounding brackets.
- `\begin{bmatrix} ... \end{bmatrix}`: Surrounds the matrix with square brackets.
- `\begin{pmatrix} ... \end{pmatrix}`: Surrounds the matrix with parentheses.
- `\begin{vmatrix} ... \end{vmatrix}`: Surrounds the matrix with single vertical bars, often used to denote determinants.
- `\begin{Vmatrix} ... \end{Vmatrix}`: Surrounds the matrix with double vertical bars, also used for determinants or norms.

Lab-5 by Alankrita

Rename the button from "Submit" to "Chin Tapak Dum Dum."

2 Introduction

This document outlines the process of modifying a "Submit" button in a mind reader application and submitting a pull request to the original GitHub repository. The repository in question is available at `https://github.com/GeekAyan/STT`. The modification includes renaming the button and fixing proportion issues.

2.1 Cloning the GitHub Repository

Step: Clone the GitHub repository using GitHub Desktop.

Action:

- Open GitHub Desktop and select `File > Clone Repository`.
- Enter the repository URL: `https://github.com/GeekAyan/STT` and select a directory to clone it.

2.2 Opening the Project in an IDE

Step: Open the cloned project using your preferred IDE (e.g., VS Code, PyCharm).

Action:

- Open the folder containing the cloned project.
- Review the `README.md` for instructions on how to run the project.

2.3 Install Dependencies

Step: Install any dependencies required by the project as per the `README.md` file.

Action:

- Set up the environment. If the project uses Python, create a virtual environment and install dependencies using: [language=bash] pip install -r requirements.txt
- Follow other system requirements mentioned in the `README.md`.

2.4 Running the Application

Step: Run the application as per the instructions in `README.md`.

Action:

- Use your IDE's terminal to run the project.
- Ensure the application works as expected.

2.5 Renaming the Submit Button

Step: Rename the button from "Submit" to "Chin Tapak Dum Dum."

Action:

- Find the code section responsible for the submit button's label.
- Modify the label. For example: [language=HTML] ¡button id="submit" name="submit"¿Chin Tapak Dum Dum¡/button¿

2.6 Fixing the Button Proportion

Step: After renaming the button, analyze and adjust its proportions.

Action:

- Check the CSS properties related to the button's size, padding, and font.
- Modify the button's CSS if needed, for example:
- Save changes and re-run the application to check the button's appearance.

2.7 Testing the Application

Step: Test the application after modifying the button.

Action:

- Run the application again to verify that the button looks correct and functions properly.

2.8 Committing the Changes

Step: Commit your changes locally.

Action:

- Stage the files and commit with a descriptive message, for example: [language=bash] git commit -m "Renamed submit button and fixed proportion issue"

2.9 Pushing Changes to Your Fork

Step: Push your changes to your GitHub fork.
Action:

- If you haven't forked the repository, go to the GitHub page and fork it.
- Add the forked repository as a remote and push your changes: [language=bash] git remote add origin https://github.com/¡YourGitHubUse git push origin main

2.10 Creating a Pull Request

Step: Create a pull request to the original repository.
Action:

- Go to your fork on GitHub and create a new pull request.
- Provide a descriptive title and details of your changes.
- Submit the pull request.

2.11 Review and Approval

Step: Wait for feedback or approval. **Action**:

- Respond to feedback or requested changes, if any.
- Once approved, your changes will be merged into the original project.