1. ***Write a simple "Hello World" program in two different programming languages of your choice. Compare the structure and syntax.***

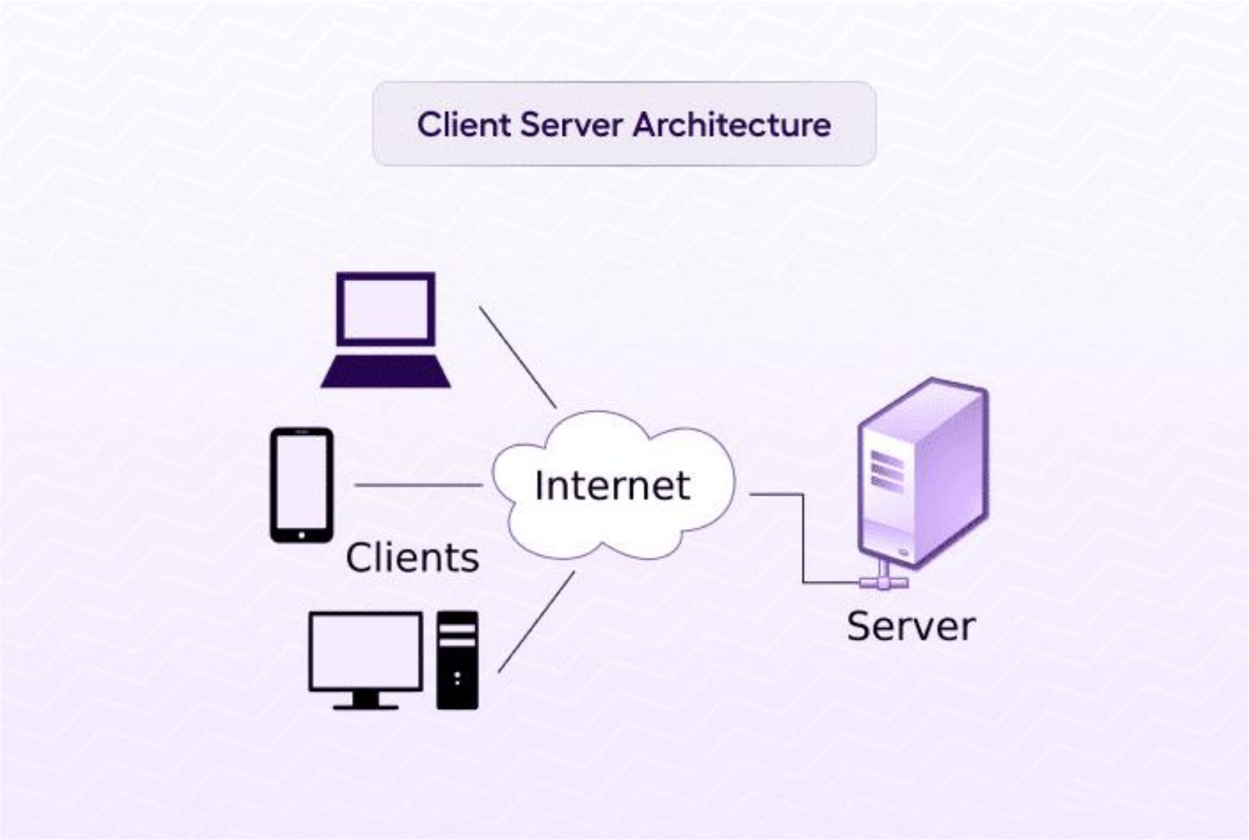
Using C Language: →

```
#include<stdio.h>

main() {

        printf("Hello World");

}
```

Using C++ Language: →

```
#include<iostream>

using namespace std;

int main(){

        cout<<"Hello World";

        return 0;

}
```

2. ***Research and create a diagram of how data is transmitted from a client to a server over the internet.***



3. Simulate HTTP and FTP requests using command line tools.

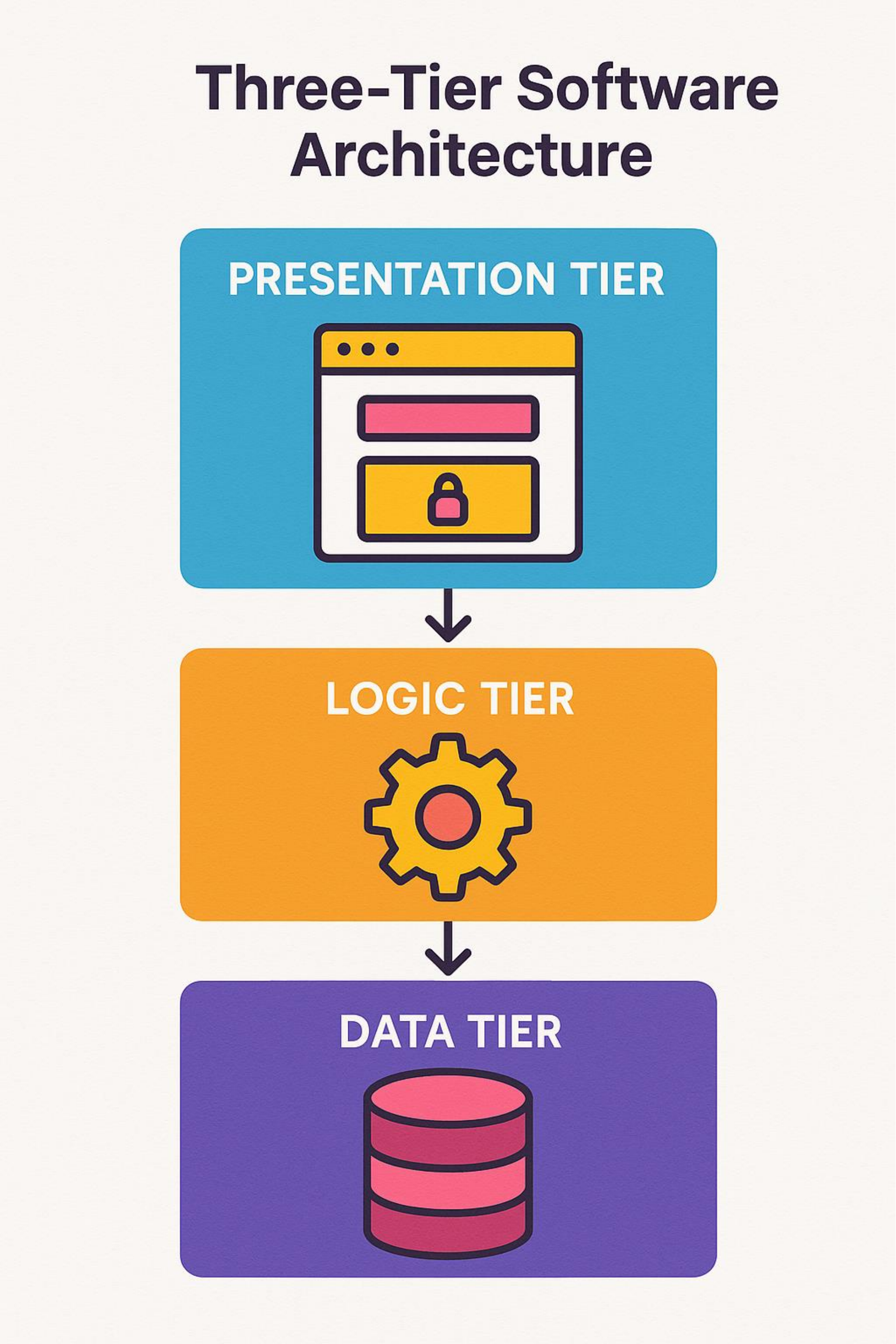
- HTTP is primarily used for web communication, and tools like curl and wget allow you to simulate requests to interact with web servers.
-
- FTP is used for file transfers, and command line tools like ftp and lftp enable you to connect to FTP servers to manage files.

4. Identify and explain three common application security vulnerabilities. Suggest possible solutions.

✅ Solutions

- **SQL Injection**
  - Use prepared statements or parameterized queries
  - Sanitize and validate all user inputs
  - Avoid dynamic SQL where possible
- **Cross-Site Scripting (XSS)**
  - Escape user input before displaying it
  - Use frameworks that auto-escape output
  - Implement Content Security Policy (CSP)
- **Broken Authentication**
  - Enforce strong password policies and use multi-factor authentication
  - Secure session handling (timeouts, secure cookies)
  - Avoid exposing session IDs in URLs

5. Design a basic three-tier software architecture diagram for a web application.



6. ***Write and upload your first source code file to GitHub.***

- Create GitHub Account: Sign up at GitHub.
- Create Repository: Set up a new repository on GitHub.
- Install Git: Download and install Git on your machine.
- Set Up Git: Configure your username and email in Git.
- Create Local Repository: Initialize a new Git repository locally.
- Add Source Code File: Create and edit your source code file.
- Stage Changes: Use `git add` to stage your file.
- Commit Changes: Commit your changes with a message.
- Link to GitHub: Add the remote repository URL.
- Push Changes: Push your local commits to GitHub.
- Verify Upload: Check your GitHub repository for the file.

7. ***Create a GitHub repository and document how to commit and push code changes.***

Creating a GitHub Repository

1. **Log In**: Access your GitHub account.
2. **New Repository**: Click the "+" icon and select "New repository."
3. **Repository Details**: Enter a name, description, and choose visibility.
4. **Create Repository**: Click "Create repository" to finalize.

Committing and Pushing Code Changes

1. **Clone Repository**: Use **git clone** to copy the repository locally.
2. **Make Changes**: Edit or create files in your local repository.
3. **Stage Changes**: Use **git add filename** to stage files.
4. **Commit Changes**: Commit with **git commit -m "Your message here"**.

5. **Push Changes**: Push to GitHub using **git push origin main**.
6. **Verify Changes**: Refresh your GitHub repository to see updates.

8. *Create a student account on Github and collaborate on a small project with a classmate.*

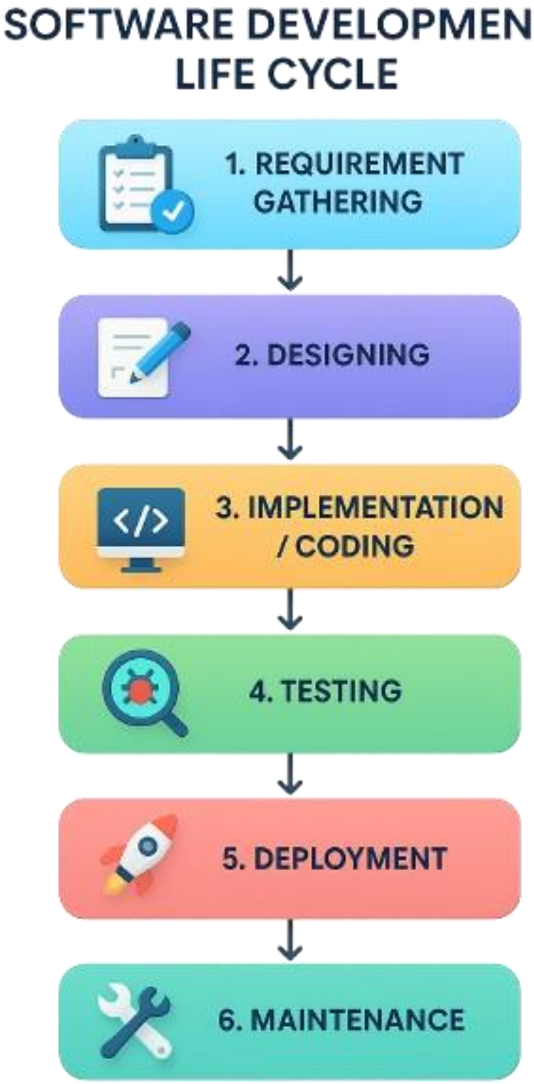
- Sign up at github.com with your email and create a new account.
- Verify your email and complete your GitHub profile setup.
- Apply for GitHub Student Pack at education.github.com.
- Create a new repository for your project from your GitHub dashboard.
- Initialize the repository with a README file and choose visibility (public/private).
- Add your classmate as a collaborator under Settings > Collaborators.
- Clone the repository to your local system using git clone <repo_url>.
- Work on the project locally and commit changes using git add., git commit -m "", and git push.
- Your classmate can pull changes, add their contributions, and push updates.
- Use GitHub issues or pull requests to manage tasks and collaborate effectively.

9. *Follow a GIT tutorial to practice cloning, branching, and merging repositories.*

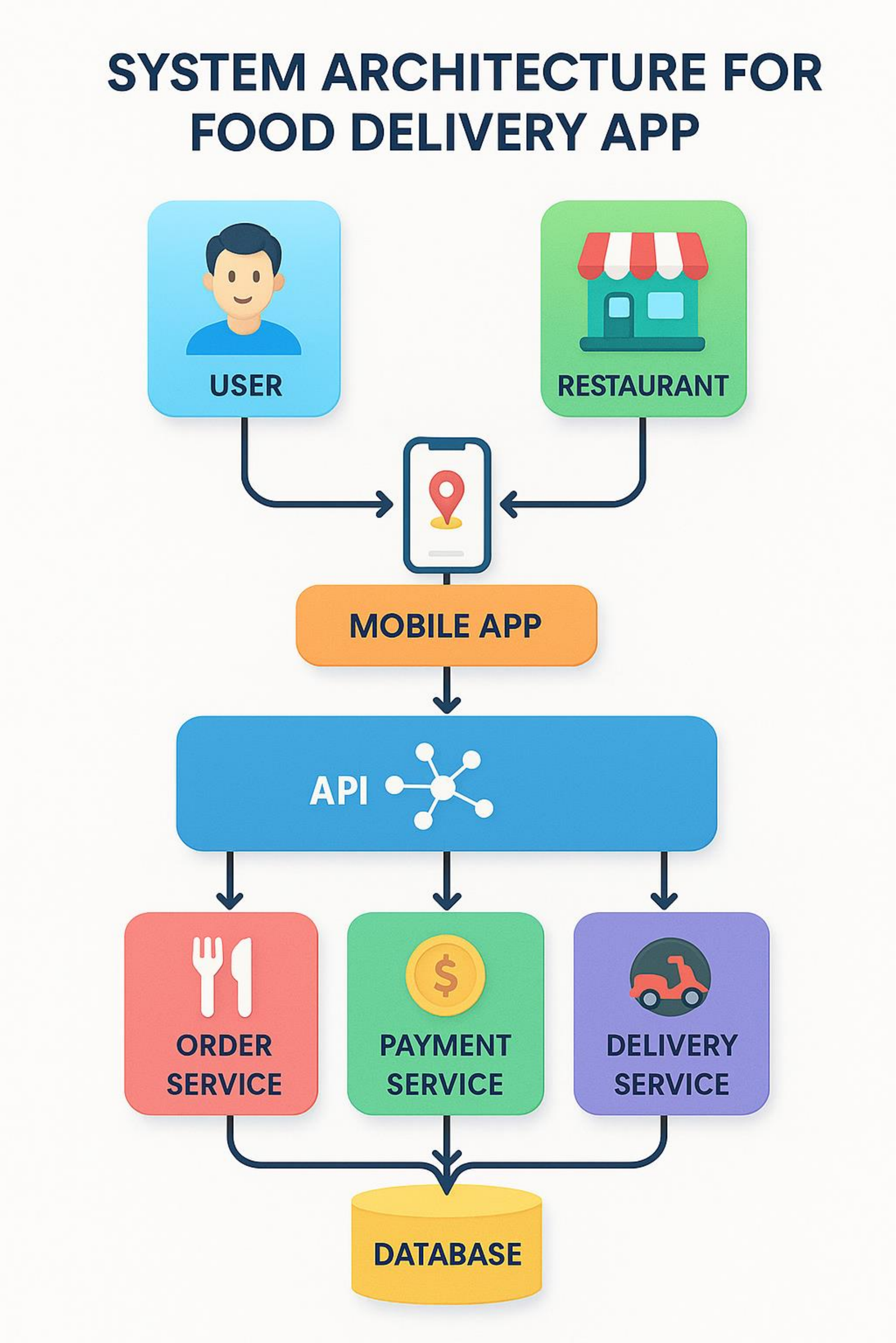
- Install Git from git-scm.com and set up your username and email using git config.
- Clone a repository using git clone <repository_URL>.

- Navigate into the project folder with cd <repo_name>.
- Create a new branch using git checkout -b <branch_name>.
- Make changes to files and stage them using git add ...
- Commit the changes using git commit -m "Your message".
- Switch back to the main branch using git checkout main.
- Merge your branch into main using git merge <branch_name>.
- Push changes to GitHub with git push origin main.
- Delete the feature branch if needed using git branch -d <branch_name>.

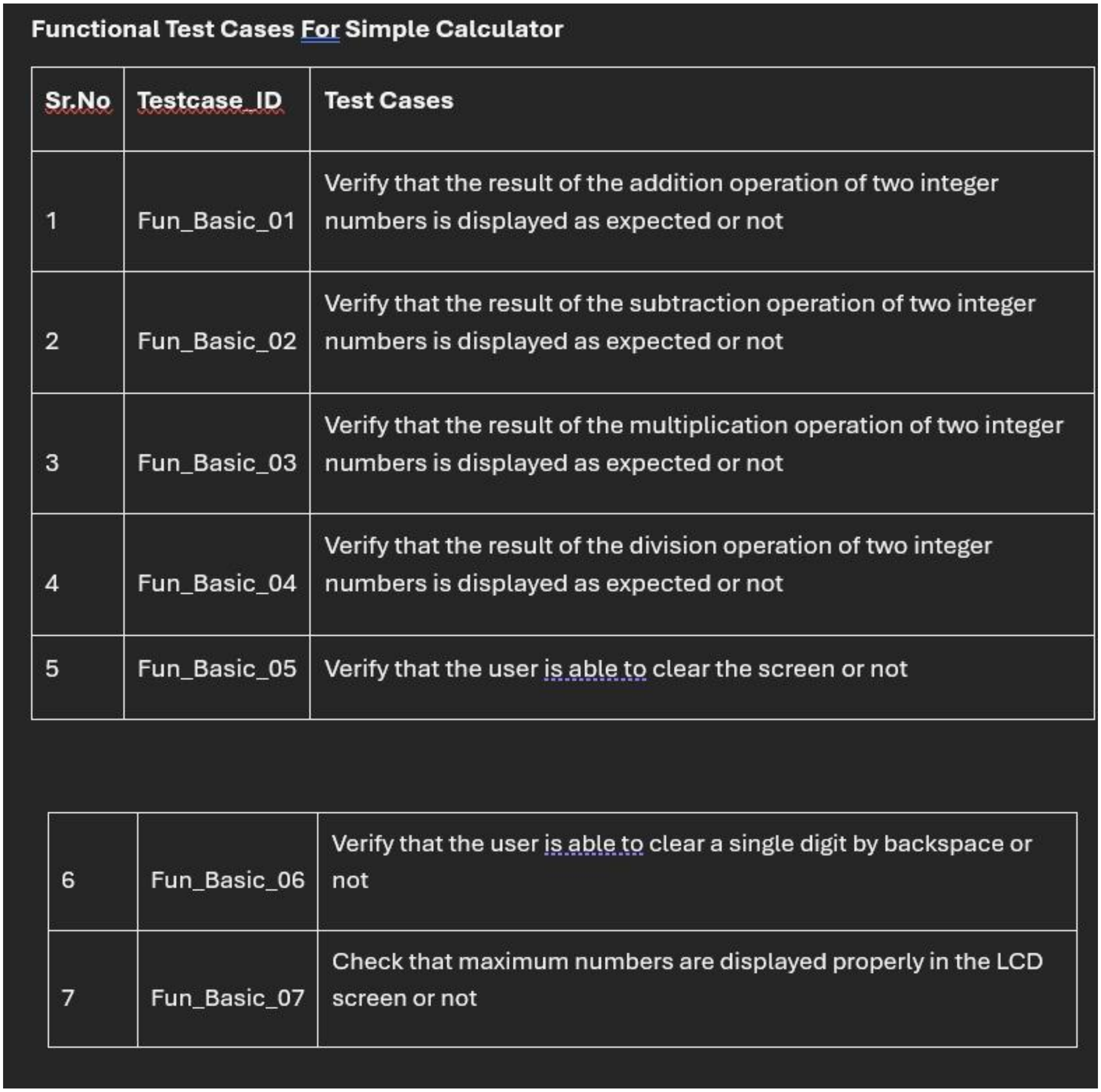
10.*Create a flowchart representing the Software Development Life Cycle (SDLC).*



11.Design a basic system architecture for a food delivery app.



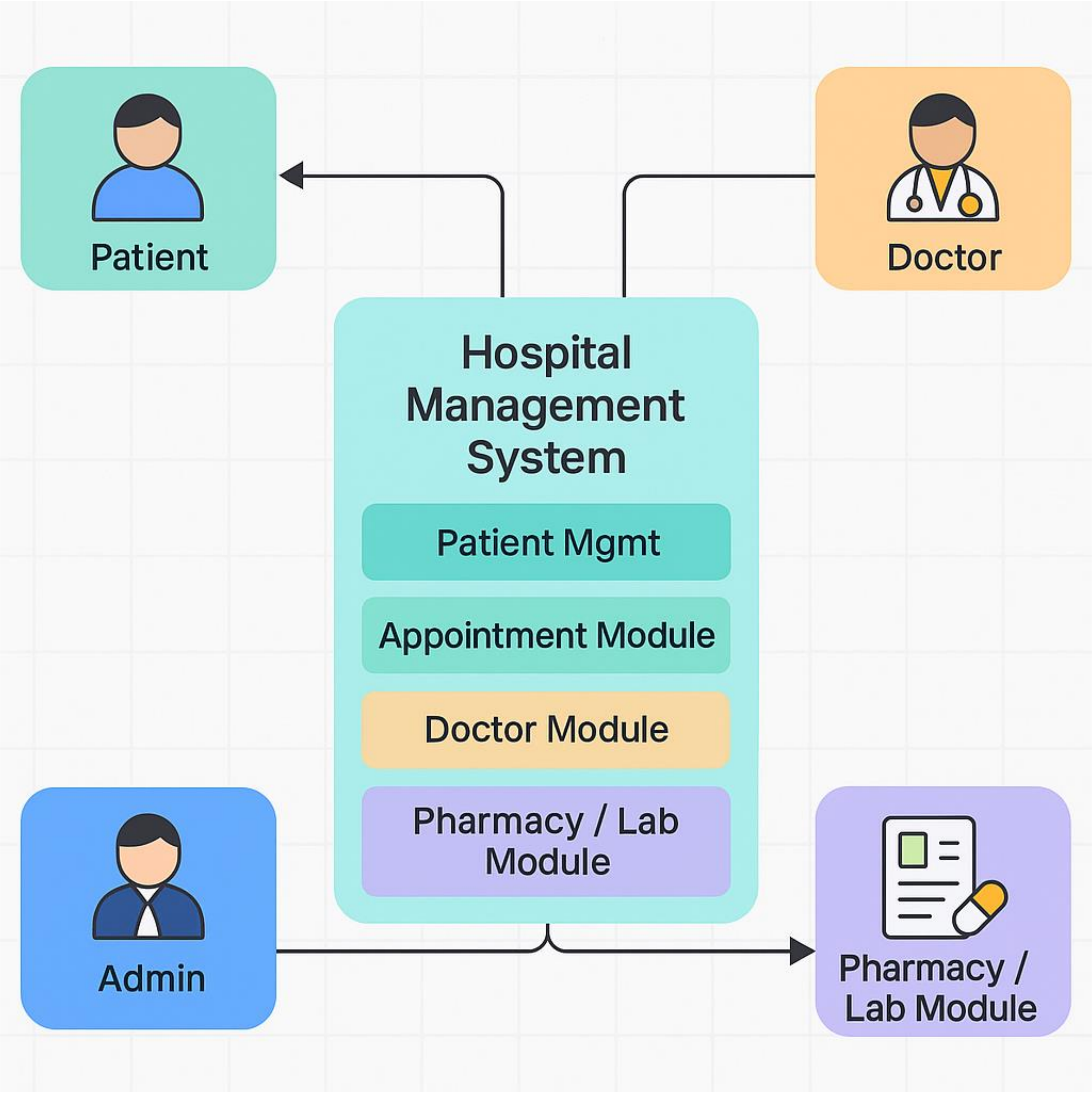
12. *Develop test cases for a simple calculator program.*



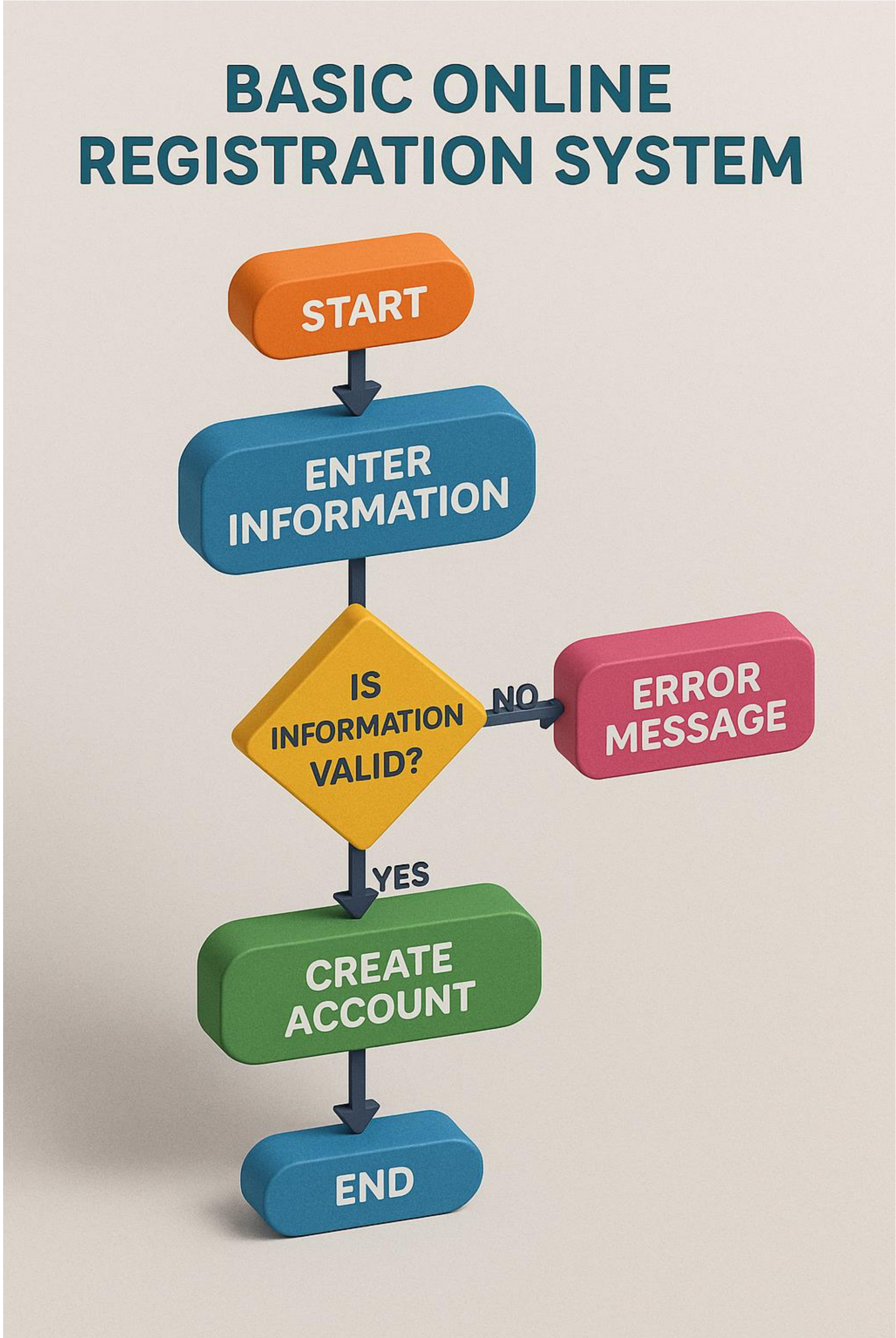
Functional Test Cases For Simple Calculator

Sr.No	Testcase_ID	Test Cases
1	Fun_Basic_01	Verify that the result of the addition operation of two integer numbers is displayed as expected or not
2	Fun_Basic_02	Verify that the result of the subtraction operation of two integer numbers is displayed as expected or not
3	Fun_Basic_03	Verify that the result of the multiplication operation of two integer numbers is displayed as expected or not
4	Fun_Basic_04	Verify that the result of the division operation of two integer numbers is displayed as expected or not
5	Fun_Basic_05	Verify that the user is able to clear the screen or not
6	Fun_Basic_06	Verify that the user is able to clear a single digit by backspace or not
7	Fun_Basic_07	Check that maximum numbers are displayed properly in the LCD screen or not

13. ***Create a DFD for a hospital management system.***



14. Draw a flowchart representing the logic of a basic online registration system.


BASIC ONLINE REGISTRATION SYSTEM
START
ENTER INFORMATION
IS INFORMATION VALID?
NO
ERROR MESSAGE
YES
CREATE ACCOUNT
END