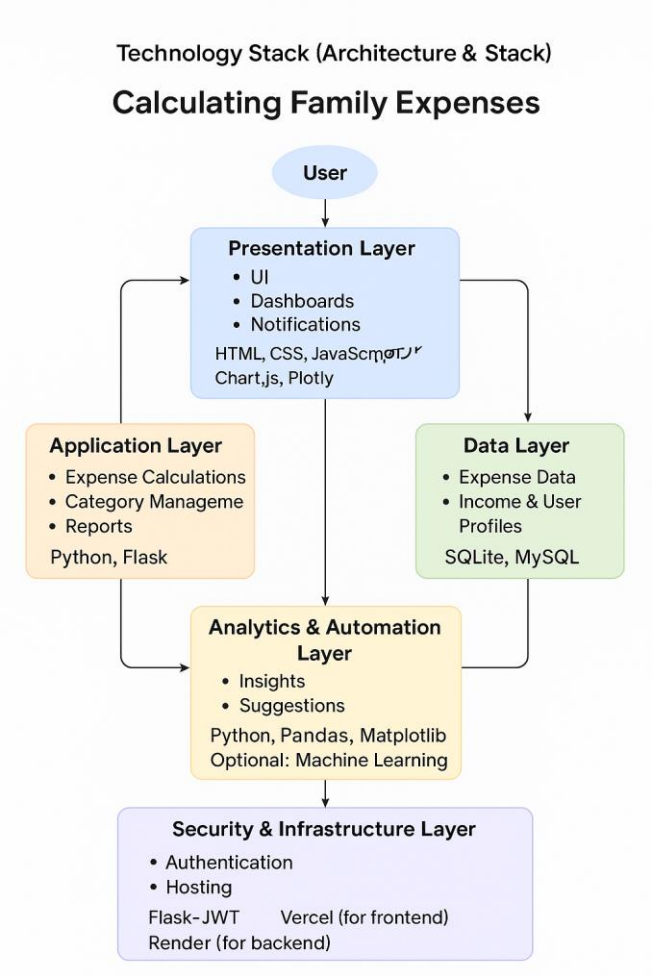
PROJECT DESIGN PHASE

TECHNOLOGY STACK(ARCHITERTURE& STACK)

Date	30/10/2025
Time id	NM2025TMID05890
Project name	Calculating Family Expenses using Service Now
Maximum mark	4 marks

Definition – Technology Stack (Architecture & Stack)

The Technology Stack refers to the set of technologies, tools, and frameworks used to design, develop, and deploy the *Calculating Family Expenses* system.
It includes both the architecture (the structural design of how different components interact) and the stack (the specific technologies used in each layer).
This combination ensures that the application is efficient, scalable, secure, and user-friendly for managing and analyzing family expenses.



Technology Stack – Components and Technologies

Component	Technology Used	Description
Frontend (Client Side)	**HTML, CSS, JavaScript**	**Builds the user interface for inputting, viewing, and analyzing expenses.**
Backend (Server Side)	**Flask (Python)**	**Manages business logic, APIs, and communication between the UI and database.**
Database	**SQLite / MySQL**	**Stores user details, expense records, categories, and reports securely.**

Component	Technology Used	Description
Authentication & Security	JWT (JSON Web Token)	Provides secure user login, session management, and data protection.
Data Visualization	Chart.js / Plotly	Displays income vs. expenses and category breakdown through visual charts.
Email / Notification Service	Flask-Mail or SMTP	Sends alerts, monthly summaries, or spending limit notifications to users.
Hosting / Deployment	Vercel (Frontend), Render (Backend)	Ensures smooth deployment and accessibility of both client and server apps.
Version Control	Git & GitHub	Used for code management, collaboration, and version tracking.

Application Characteristics :

S.No	Characteristics	Description	Technology
1.	Open-Source Frameworks	Utilizes open-source technologies for flexibility and cost efficiency.	HTML, CSS, JavaScript, Flask (Python)
2.	Security Implementations	Implements user authentication, role-based access control, and encrypted data storage.	JWT Authentication, Flask Security, SSL
3.	Scalable Architecture	Designed to support multiple families and large datasets with minimal latency.	Flask REST API, SQLAlchemy, Cloud Hosting (Render)
4.	Availability	Application hosted on cloud platforms ensuring high uptime and reliability.	Render, Vercel, Cloud Database
5.	Performance	Optimized for quick data retrieval, smooth dashboard loading, and fast expense calculations.	Indexed Database Tables, Caching, Optimized Queries