**System Architecture**

FindMyCollege adopts a 3-Tier Architecture design, namely:

|  |  |  |
| --- | --- | --- |
| **Presentation Layer** | **Application Layer** | **Data Layer** |
| 1.     HTML  2.     CSS  3.     JavaScript | 1.     Python  2.     Flask | 1. SQLAlchemy |

This tiered architecture means that each layer is independent of the other layers. Hence, each layer can be modified or scaled without affecting the system’s performance, therefore also promotes modularity and increases separation of concerns. Furthermore, having the data layer separated from the other layers adds security to the system, making it difficult for any user to gain unwanted access to the database with everyone’s private information.

We also adhered to good software engineering principles of

1.     Good design

a.     Reusability of features

§  Google Charts API is reusable to make any graphs the programmer needs.

§  Methods for user account creation, the authentication process and storing of user data in a database can be reused for another project.

b.     Testability of features

§  Each feature can be tested independently.

§  Features like signup, personality test page, storing of data in the database can all be tested separately.

c.     Maintainability of features

§  Ensured readability of code

§  Included comments to help anyone else reading the source code understand what each function does.

d.     Extensibility of features

§  The flexible architecture allows new modules or components to be added without affecting the existing system.

2.     Design Patterns

a.     Single responsibility principle

§  Our application is broken down into smaller, individual functions, each with a single responsibility.

§  Improves reusability as each function can be reused in other modules.

§  Improves maintainability as each function is less likely to affect other functions.

3.     Design Principles

a.     High cohesion

§  The modules in the system have a single, well-defined responsibility.

b.     Low coupling

§  Changes in one module will not affect other functions.

§  Eg. In take\_test and Login modules, the former facilitates displaying test questions and storing of user data while the latter facilitates user authentication. These modules do not have overlapping responsibilities and does not affect each other at all.

c.     Open-Closed

§  Existing code allows user data (email, password, website-specific data) to be used simultaneously.

§  Eg. The code needs no further modification when trying to get any user data. (Closedness)

§  Extension of existing classes is possible.

§  Eg. Website-specific data can be modified to store whatever the programmer requires. (Openedness)

d.     Separation of concerns

§  The application consists of independent modular component, having adopted a 3-tier architecture.

§  The code is more readable since all components do not have overlapping functionalities.

Overall, a tiered architecture and good software design principles in FindMyCollege provides benefits like scalability, modularity, maintainability, reusability and security.