**Project Design Phase-II**

**Technology Stack (Architecture & Stack)**

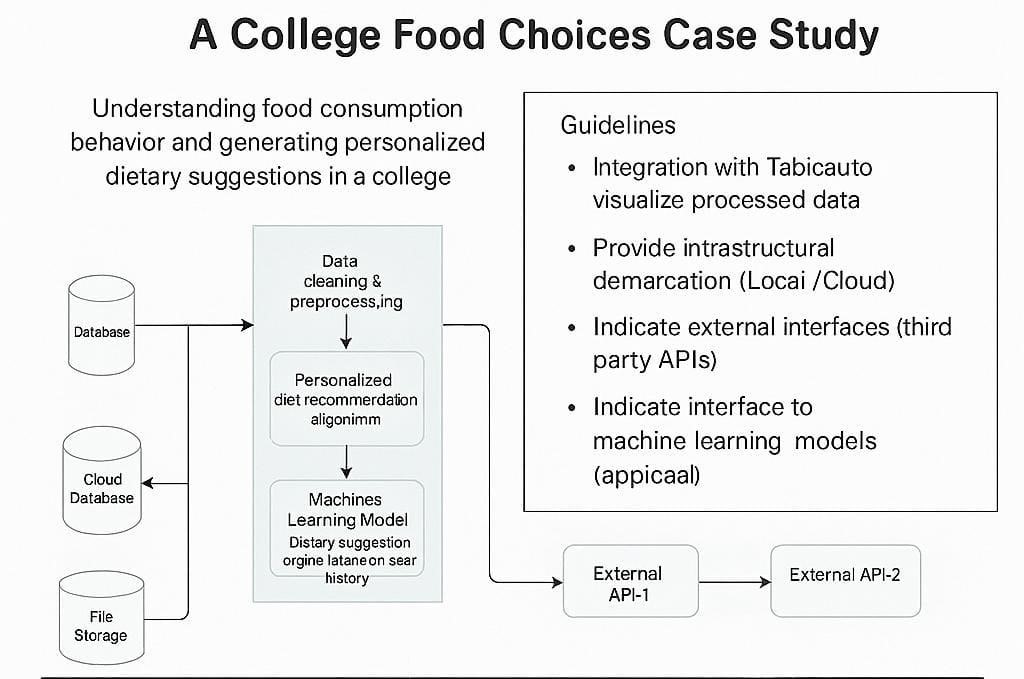
|  |  |
| --- | --- |
| Date | 03 July 2025 |
| Team ID | LTVIP2025TMID51397 |
| Project Name | Comprehensive Analysis and Dietary  Strategies with Tableau: A College Food  Choices Case Study |
| Maximum Marks | 4 Marks |

**Technical Architecture:**

The Deliverable shall include the architectural diagram as below and the information as per the table1 & table 2

**Example: A College Food Choices Case Study**

**Reference:** [**https://developer.ibm.com/patterns/ai-powered-backend-system-for-order-processingduring-pandemics/**](https://developer.ibm.com/patterns/ai-powered-backend-system-for-order-processing-during-pandemics/)



**Table-1 : Components & Technologies:**

|  |  |  |  |
| --- | --- | --- | --- |
| **S.No** | **Component** | **Description** | **Technology** |
| 1. | User Interface | How user interacts with application  Dashboard, Survey Input UI | Dashboard, Survey Input UI Tableau Dashboards, React.js, HTML/CSS |
| 2. | Application Logic1 | Data collection logic (surveys, cafeteria logs, manual entries) | Python scripts, Tableau Web Data Connectors |
| 3. | Application Logic2 | Data cleaning & preprocessing | Tableau Prep, Python (Pandas) |
| 4. | Application Logic3 | Personalized diet recommendation algorithm | Python (scikit-learn), ML Models |
| 5. | Database | Storage of raw & processed dietary data | MySQL, NoSQL (MongoDB) |
| 6. | Cloud Database | Cloud-based access to dietary  datasets | . Google Firebase, AWS RDS,  Snowflake |
| 7. | File Storage | Storing reports, charts, and user uploads | AWS S3, Google Drive API, Tableau Public |
| 8. | External API-1 | Nutrition data from external sources | USDA Food Data Central API |
| 9. | External API-2 | Student info or campus data access | College ERP API, Google Forms API |
| 10. | Machine Learning Model | Dietary suggestion engine based on user history | Python ML Model (KNN, Decision Tree) |
| 11. | Infrastructure  (Server / Cloud) | Hosting Tableau server or cloud dashboards | Tableau Server, AWS EC2, Google Cloud. |

**Table-2: Application Characteristics:**

|  |  |  |  |
| --- | --- | --- | --- |
| **S.No** | **Characteristics** | **Description** | **Technology** |
| 1. | Open-Source Frameworks | List the open-source frameworks used | Python (Pandas, NumPy, scikitlearn), MySQL |
| 2. | Security  Implementations | .Access control for student health data,  APIs, and dashboards | OAuth 2.0, Encryption (SHA256), IAM Roles |
| 3. | Scalable  Architecture | Modular layers: UI – Processing – Storage – ML – Visualization | Microservices, Docker, Tableau Extensions |
| 4. | Availability | Hosted on cloud with dashboard backup, load-balanced APIs | Tableau Online, Load Balancer (AWS/GCP) |
| 5. | Performance | Fast dashboard loading, efficient ML model execution, data caching | Tableau Extracts, CDN, Redis (optional) |

**References:**

[**https://c4model.com/**](https://c4model.com/) [**https://developer.ibm.com/patterns/online-order-processing-system-during-pandemic/**](https://developer.ibm.com/patterns/online-order-processing-system-during-pandemic/) [**https://www.ibm.com/cloud/architecture**](https://www.ibm.com/cloud/architecture) [**https://aws.amazon.com/architecture**](https://aws.amazon.com/architecture)

[**https://medium.com/the-internal-startup/how-to-draw-useful-technical-architecture-diagrams2d20c9fda90d**](https://medium.com/the-internal-startup/how-to-draw-useful-technical-architecture-diagrams-2d20c9fda90d)