

Week- 3

Team 38 – Patient Insurance & Management System

Final Technology stack

Backend: Python

Why?: Python offers a vast ecosystem of libraries and frameworks, making it easier to integrate with various services and data manipulation tasks. Its readability and maintainability make it ideal for rapid development.

Web Framework: Flask

Why?: Flask provides the flexibility and simplicity needed for quick development and prototyping. It's lighter than Django and offers more control over how you structure your application.

Frontend: React

Why?: React allows for a component-based architecture, making it easier to reuse code and manage states across large applications. It is widely adopted and has a strong community and ecosystem.

API: GraphQL or REST API (TBD)

Why?:

GraphQL: Offers greater flexibility and efficiency in data retrieval, essential for complex healthcare records. It also supports real-time updates.

REST: Provides simplicity and aligns well with HTTP protocols, making it easier to understand and implement.

Database: SQL or NoSQL (TBD)

Why?:

SQL: Robust and ACID compliant, ideal for complex queries and transactions that are common in healthcare systems.

NoSQL: Scales horizontally and is schema-less, which can be beneficial for storing semi-structured or unstructured data, such as patient logs or real-time data.

Note: The decision between GraphQL and REST, as well as SQL and NoSQL, will be finalized based on the specific data requirements and interaction complexity as we proceed with the project sprints.