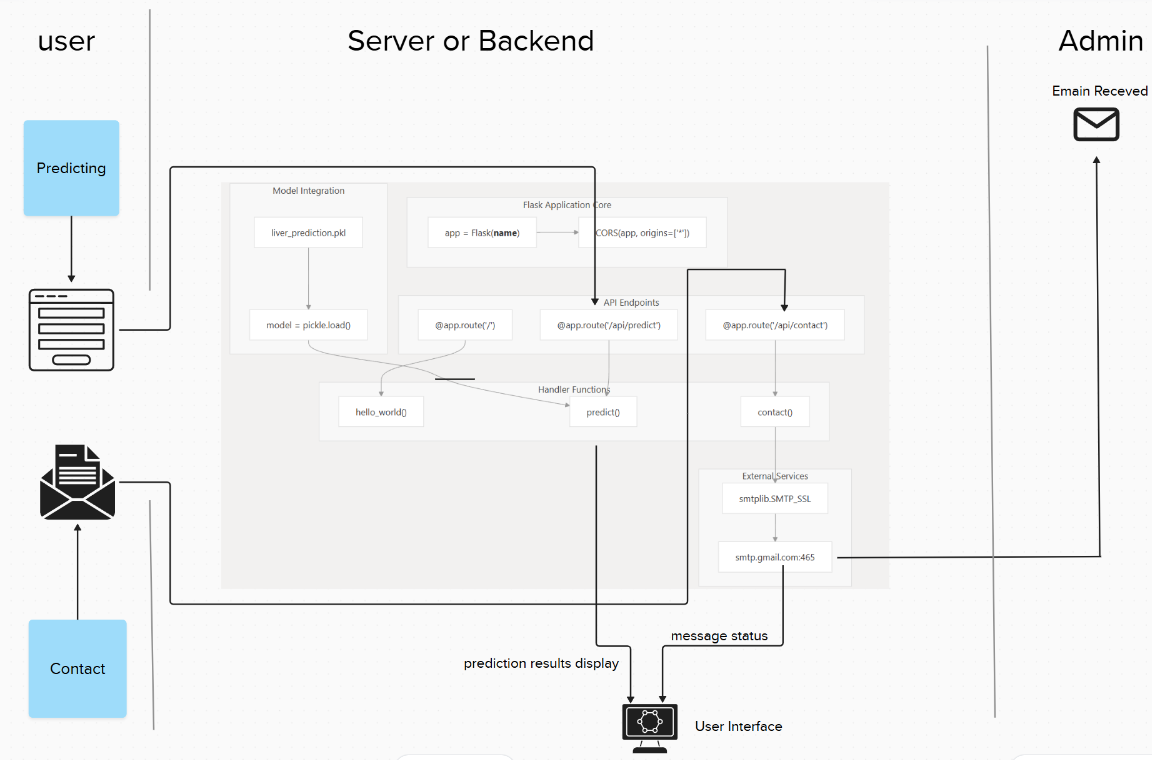
**Project Design Phase-II**

**Technology Stack (Architecture & Stack)**

|  |  |
| --- | --- |
| Date | 16 June 2025 |
| Team ID | LTVIP2025TMID35624 |
| Project Name | Revolutionizing Liver Care: Predicting Liver Cirrhosis Using Advanced Machine Learning Techniques |
| Maximum Marks | 4 Marks |

**Technical Architecture:**

****

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

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

|  |  |  |  |
| --- | --- | --- | --- |
| **S.No** | **Component** | **Description** | **Technology** |
|  | User Interface | Web-based UI for patients and doctors to enter data and view results and contact with admin | HTML, CSS, JavaScript |
|  | Application Logic-1 | Backend logic to manage routing, form handling, API endpoints | Python (Flask) |
|  | Application Logic-2 | Contact form handler that sends messages to admin via Gmail SMTP | Python smtplib, email.message |
|  | Application Logic-3 | ML model prediction handler: Preprocess data, predict, return response | Flask, NumPy, Pandas |
|  | External API-1 | Not applicable in this version | - |
|  | External API-2 | Gmail API used to send user messages to admin | Google Gmail SMTP |
|  | Machine Learning Model | Predict liver cirrhosis using health data | Random Forest Classifier (scikit-learn) |
|  | Infrastructure (Server / Cloud) | App deployed on local server and cloud (Docker-ready) | Localhost / Flask / Docker-ready |

**Table-2: Application Characteristics:**

| **S.No** | **Characteristics** | **Description** | **Technology** |
| --- | --- | --- | --- |
|  | Open-Source Frameworks | Used frameworks and libraries are open-source | Flask, scikit-learn, Pandas, NumPy, Matplotlib & Seaborn |
|  | Security Implementations | Input validation, secure SMTP mail handling, CORS headers in backend | Gmail Auth, Flask-CORS. |
|  | Scalable Architecture | Backend is modular, ML model is containerizable and API is RESTful | 3-tier architecture (Frontend, API, Model) |
|  | Availability | Flask API deployable with Docker support ensures availability in onrender.com server  Html/Css deployable in netlify.app | Docker |
|  | Performance | Optimized preprocessing and fast inference (<2 sec); prediction via local model | Flask API, pickle load, low latency model |

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