## Detailed requirements

**Functional Requirements:**

1. **User Requirements**

* Users should be able to upload high-resolution skin lesion images in common formats (e.g., JPEG, PNG).
* Users should have the ability to review uploaded images before proceeding with analysis.
* The system must notify users if uploaded images do not meet quality standards (e.g., resolution or format issues).

1. **Image Preprocessing**

The system will preprocess uploaded images by performing the following:

* **Contrast Enhancement**: Improve image quality by adjusting brightness and contrast.
* **Resizing**: Resize images to the required input dimensions of the deep learning model.
* **Normalization**: Standardize pixel intensity values for optimal model performance.
* **Segmentation**: Automatically isolate skin lesions from the background using advanced image processing techniques.

1. **Disease Detection**

* The software will use both traditional machine learning and deep learning models for disease detection.
* Employ a **Convolutional Neural Network (CNN)** (e.g., Resnet50, InceptionV3, or a custom-designed CNN) for the classification of skin lesions.
* The system will distinguish between multiple skin disease types such as Benign lesions, Malignant lesions, Melanoma, Basal cell carcinoma and Squamous cell carcinoma.

1. **Result Display**

Display the following results to the user after analysis:

* Detected disease name (if any).
* Confidence score (for example "85% confidence this is melanoma")
* Visualized results, such as bounding boxes or heatmaps highlighting the lesion in the uploaded image.
* Offer **an advice meeting with a doctor** based on detected disease.
* Provide a summary report in a user-friendly format.

1. **Data Management**

* Securely save patient data, including uploaded images, analysis results, and reports, for future reference.
* Enable users to view and manage their history of previous reports.
* Provide an option to export reports in formats such as **PDF**.

**Non-Functional Requirements:**

1. **Performance**

* The model should analyse and return results within 5 seconds for each uploaded image.

1. **Scalability**

* The system should handle up to 1,000 concurrent users during peak hours without degradation in performance.

1. **Accuracy**

* The detection model should achieve at least 85% accuracy, and 80% sensitivity based on test dataset evaluation.

1. **Security**

* User data, including uploaded images and diagnosis results, must be securely stored using encryption.

1. **Reliability**

* The system should have an uptime of 99.9% and handle unexpected errors with user-friendly messages.

**Technical Requirements:**

1. **Programming Languages**

* **Python**: For deep learning model development, data preprocessing, and backend integration.
* **JavaScript/TypeScript**: For building a responsive and dynamic frontend.

1. **Frameworks and Libraries**

* **Deep Learning:** TensorFlow or PyTorch for model development and training.
* **Backend**: Django or Flask for building RESTful APIs.
* **Frontend:** React or Angular for creating an intuitive user interface.
* **Data Visualization:** Matplotlib, Seaborn, or Plotly for presenting model results and metrics graphically.

1. **Database**

* **SQL**: MySQL for structured data storage (e.g., patient records).
* **NoSQL**: Firebase Realtime Database or MongoDB for unstructured or semi-structured data (e.g., image metadata, logs).

1. **Development Tools**
   * + **Image Processing**: OpenCV for preprocessing tasks like resizing, normalization, and segmentation.

* **Integrated Development Environment (IDE)**: VS Code, PyCharm, or Jupyter Notebooks for coding and testing.
* **Version Control**: GitHub for collaborative development and versioning.

**Stakeholders**

* + Patients: Individuals who will upload skin lesion images and view diagnostic results.
  + Healthcare Providers: Dermatologists or clinicians who may use the software to assist in diagnosing skin diseases.
  + Researchers: Researchers may use the system for analysing data trends or improving diagnostic algorithms.