BodyScan App: AI-Powered Medical Imaging Diagnostics Report

1. Project Overview

The **BodyScan App** is a professional-grade medical imaging application designed to assist healthcare professionals in the preliminary diagnosis of critical conditions. It leverages state-of-the-art deep learning models to provide a unified interface for analyzing various types of medical scans, including X-rays and MRIs. The application aims to enhance diagnostic accuracy and efficiency in clinical settings.

2. Key Features

The BodyScan App is built with a modular architecture, supporting four specialized diagnostic modules:

Module	Diagnostic Focus	Modality
Chest X-ray	Pneumonia Detection	X-ray
Brain Tumor	Tumor Identification	MRI/CT
Diabetic Retinopathy	Retinal Analysis	Fundus Photography
Fracture Detection	Bone Fracture Identification	X-ray

3. System Architecture

The project employs a modern **Client-Server Architecture** to ensure scalability and maintain a clear separation of concerns:

1. **Frontend (Java/JavaFX):** A cross-platform desktop application that offers an intuitive user interface for uploading images and visualizing diagnostic results.

2. **Backend (Python/Flask):** A RESTful API service responsible for handling complex image processing tasks and performing deep learning inference using TensorFlow/Keras.
3. **AI Engine:** Comprises pre-trained Convolutional Neural Networks (CNNs) specifically optimized for various medical image classification tasks.

4. Project Structure

The `BodyScan_App` repository is structured as follows:

Directory	Description
`ai_models/`	Contains the Python backend and AI models, including `app.py` (unified Flask API service), `requirements.txt` (Python dependencies), and `APIs/` (legacy individual API scripts).
`src/`	Houses the Java frontend source code, with `BodyScan/` containing the JavaFX application logic.
`presentation/`	Stores project documentation and slides, such as `BodyScan AI for Medical Imaging.pdf`.
`.gitignore`	Specifies files and directories to be ignored by Git.
`README.md`	Provides an overview and documentation for the project.

5. Getting Started

Backend Setup (Python)

Prerequisites: Python 3.8+ installed.

```
# Navigate to the backend directory
cd ai_models

# Install dependencies
pip install -r requirements.txt

# Start the API service
python app.py
```

The backend service will be accessible at `http://localhost:5000`. Its health status can be checked at `http://localhost:5000/health`.

Frontend Setup (Java)

Prerequisites:

- Java Development Kit (JDK) 11 or higher.
- JavaFX SDK.

Running the App:

1. Import the project into a suitable IDE (e.g., IntelliJ IDEA, Eclipse, or VS Code).
2. Ensure that the JavaFX libraries are correctly added to your project's build path.
3. Execute `src/BodyScan/Main.java` to launch the application.

6. API Documentation

The backend provides a unified endpoint for all diagnostic modules:

- **Endpoint:** `POST /predict/<model_type>`
- **URL Parameters:** `model_type` (can be `chest`, `brain`, `diabetic`, or `fracture`)
- **Body:** `multipart/form-data` with a `file` field containing the medical image.

Example Response:

```json
{
  "model": "chest",
  "prediction": "Pneumonia",
  "confidence": 98.5,
  "raw_score": 0.985
}
```

7. License

This project is licensed under the MIT License. Further details can be found in the `LICENSE` file within the repository.

8. Conclusion

The BodyScan App represents a significant advancement in AI-powered medical imaging diagnostics. Its modular design, robust client-server architecture, and specialized deep learning models for various conditions offer a powerful tool for healthcare professionals. The project's clear structure and comprehensive documentation facilitate ease of setup and use, contributing to improved diagnostic workflows and patient care.