PROJECT REPORT

Technologies used

| **🔧 Category** | **🛠️ Technology** | **💡 Purpose** |
| --- | --- | --- |
| **Programming Language** | Python, JavaScript | Backend & frontend development |
| **Backend Framework** | Flask | Serve API, handle authentication, model prediction |
| **Frontend Framework** | React.js | Build interactive UI |
| **Styling** | Tailwind CSS | Utility-first modern styling |
| **Machine Learning** | TensorFlow / PyTorch / Sklearn | Train and serve medical image classifier |
| **Image Processing** | OpenCV / PIL | Image loading, preprocessing |
| **HTTP Client (Frontend)** | Axios | Send requests from React to Flask backend |
| **Authentication** | Flask-JWT-Extended, JWT | User login, signup, session management |
| **Database** | MongoDB | Store user and prediction data |
| **ORM / DB Client** | PyMongo | Interact with MongoDB |
| **Routing (Frontend)** | React Router DOM | Navigate between login, dashboard, etc. |
| **State Management** | React Hooks (useState, useEffect) | Manage UI state |
| **API Testing Tool** | Postman | Test Flask backend endpoints |
| **Version Control** | Git & GitHub | Source code tracking and hosting |
| **Editor / IDE** | VS Code / PyCharm | Development environment |
| **Deployment** | Render / Netlify / Vercel | Host backend and frontend online |
| **Environment Config** | .env file | Store secret keys, DB URI, etc. |
| **Documentation** | README.md, Report (PDF) | Project explanation and instructions |