**Project Design Phase-II**

**Technology Stack (Architecture & Stack)**

|  |  |
| --- | --- |
| Date | 26/06/2025 |
| Team ID | LTVIP2025TMID33042 |
| Project Name | HematoVision-Blood Cell Classifier |
| Maximum Marks | 4 Marks |

**Overview-** Identifies the stack for UI, image processing, ML inference, reporting, packaging, and storage.

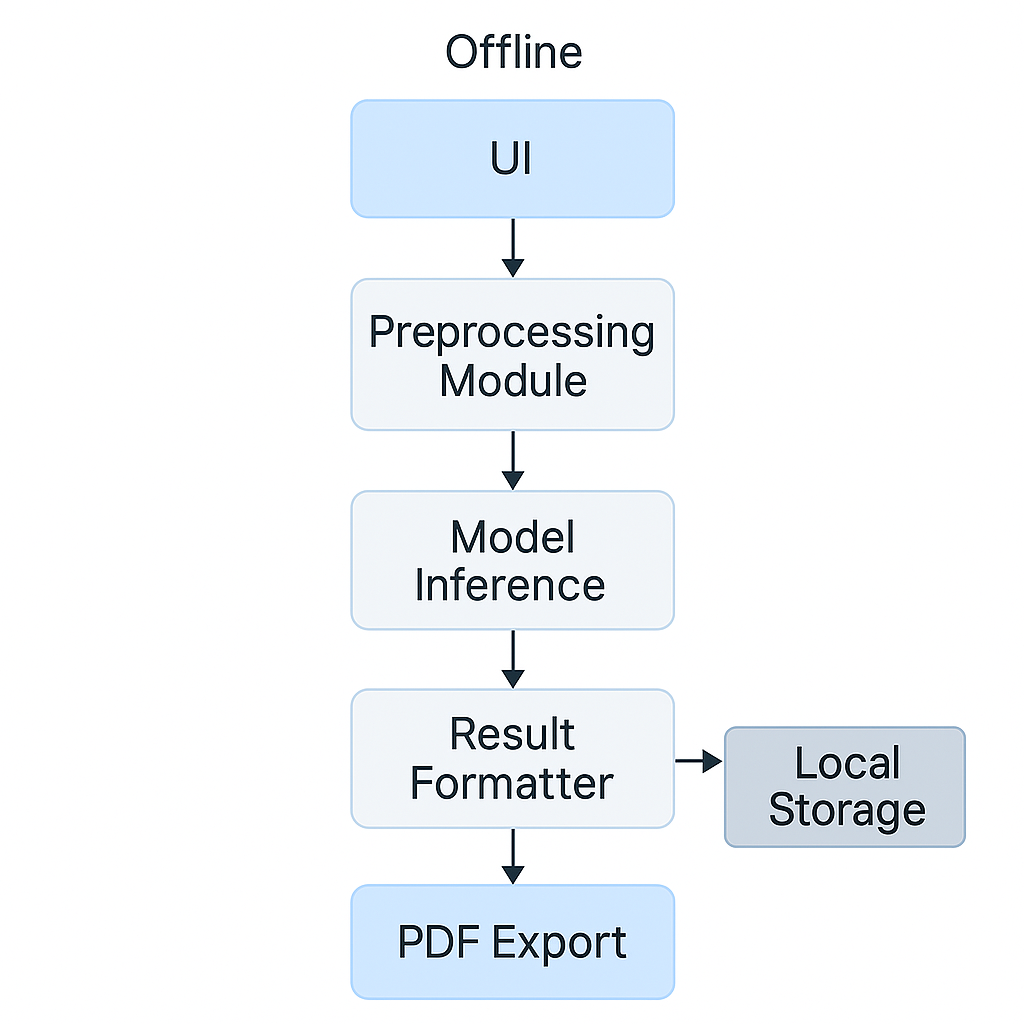
• Diagrams the component architecture so you can see how modules plug together in an offline bundle.

**Table-1: Components & Technologies**

| **S.No** | **Component** | **Description** | **Technology / Tool** |
| --- | --- | --- | --- |
| 1 | User Interface | Local web UI for uploading images, viewing results & reports | Flask, HTML5, CSS3, Bootstrap, JavaScript |
| 2 | Application Logic-1 | Image preprocessing (resizing, normalization, filtering) | Python, OpenCV, scikit-image |
| 3 | Application Logic-2 | Model inference pipeline | Python, TensorFlow/Keras |
| 4 | Application Logic-3 | Explanation module (heatmaps, confidence bars, textual rationale) | Python, Grad-CAM, Matplotlib |
| 5 | Database | Store user settings & session metadata | SQLite or JSON flat-files |
| 6 | Cloud Database | — | Not used (fully offline) |
| 7 | File Storage | Persist uploaded images & generated PDF reports | Local filesystem (OS directories) |
| 8 | External API-1 | — | Not used |
| 9 | External API-2 | — | Not used |
| 10 | Machine Learning Model | Convolutional Neural Network for blood cell classification | Custom CNN (TensorFlow), quantized ONNX |
| 11 | Infrastructure | Offline packaging & deployment | PyInstaller, custom .bat launcher |

**Table-2: Application Characteristics**

| S.No | Characteristic | Description | Technology / Approach |
| --- | --- | --- | --- |
| 1 | Open-Source Frameworks | All core components built on free, community-driven projects | Flask, TensorFlow, OpenCV, ReportLab, PyInstaller |
| 2 | Security | Ensures no data leaves the user’s machine; sandboxed file reads/writes | Offline-only operation; OS-level file perms |
| 3 | Scalable Architecture | Modular design lets you swap or extend preprocessing, inference, reporting | Python modules + REST endpoints |
| 4 | Availability | Runs without network; packaged dependencies guarantee “cold start” readiness | Local bundle via PyInstaller; splash screen |
| 5 | Performance | Inference under 5 s on 512×512 inputs; caches pre-processed intermediates | Model quantization; in-memory caching |

****