HematoVision – Blood Cell Classification

TeamID:LTVIP2026TMIDS47037
Team Size: 4

Team Members

- **Team Leader:** C.ArunaKumari – Project Lead & Frontend Developer
- **Team Member:** D.Anusha – Backend Developer (Model Training & Integration)
- **Team Member:** K.Leela Shiva Maruthi Reddy – Database & Cloud Specialist (MongoDB Atlas, Deployment)
- **Team Member:** B.Deepthi – Documentation & Testing Analyst

---

1. Introduction

Project Title

HematoVision – Blood Cell Classification

Purpose

The purpose of this project is to **automate blood cell classification using Artificial Intelligence (AI)**. The system classifies uploaded microscope blood images into **Red Blood Cells (RBCs), White Blood Cells(WBCs),Platelets**.
It also detects **abnormal or infected cells** that could indicate diseases such as anemia or leukemia.

This project aims to assist **hematologists and lab technicians** in reducing manual workload, improving diagnostic speed, and minimizing errors in blood cell identification.

---

2. Project Overview

Objective

- Automate blood cell classification using a trained AI model.
- Provide real-time and accurate diagnostic results.
- Enable doctors to view and download patient reports instantly.
- Reduce manual human effort in routine lab processes.

Features

- Upload blood smear images for analysis.
- AI-based automatic cell classification.
- Abnormality detection and flagging.
- Real-time result visualization and reporting.
- Secure login and role-based access (Admin, Doctor, Technician).
- Cloud-based storage for reports and images.

---

3. Architecture

Frontend

- Built using **React.js**, **HTML5**, and **CSS3**.
- Provides modules for login, registration, image upload, and report viewing.
- Offers dashboards for doctors and lab technicians with visual charts and summaries.
- Designed for accessibility on both desktop and mobile screens.

Backend

- Developed using **Node.js** and **Express.js** for managing RESTful APIs.
- Integrates with **Python (Flask/FastAPI)** for AI model inference.
- Handles image uploads, user management, and report generation.
- Includes secure API endpoints for data communication.

Database

- Uses **MongoDB Atlas** as a NoSQL database for scalability and cloud accessibility.
- Stores user information, patient details, image metadata, and classification reports.
- Ensures data integrity and privacy with role-based access control.

---

4. Setup Instructions

Prerequisites

- Node.js and npm installed
- MongoDB Atlas account
- Python environment with TensorFlow or PyTorch installed
- React.js environment setup

Installation Steps

1. Clone the HematoVision repository from GitHub.
2. Run npm install to install dependencies.
3. Set up MongoDB Atlas credentials in .env.
4. Start backend: node server.js.
5. Start frontend: npm start.
6. Upload sample blood cell images to test the system.

---

5. Folder Structure

HematoVision/

```
│
├── frontend/
│   ├── components/
│   ├── pages/
│   ├── CSS/
│   └── App.js
│
├── backend/
```

```
│   ├── routes/
│   ├── models/
│   ├── controllers/
│   └── server.js
│
├── AI_Model/
│   ├── model.py
│   └── preprocess.py
│
├── database/
│   └── MongoDB Atlas
│
└── README.md
```

This structure ensures **scalability**, **modularity**, and **clean code management**.

---

6. Running the Application

Once all configurations are complete:

- Start backend and frontend servers.
- Access the web app(or hosted URL).
- Upload a blood smear image.
- The AI model processes the image and generates classification results.
- Results and reports are stored automatically in MongoDB Atlas.

No external server setup is required beyond initial deployment.

---

7. API Documentation

Image Upload API

Endpoint: POST /api/upload
Purpose: Upload blood cell images for classification
Authentication: Required
Response: JSON with classification results and probability values

---

Classification API

Endpoint: POST /api/classify
Purpose: Send image data to AI model for classification
Method: POST
Response Example:

```
{
 "RBC": 60,
 "WBC": 30,
 "Platelets": 10,
 "Abnormal_Cells": 2
}
```

Report Generation API

Endpoint: GET /api/reports/:patientId
Purpose: Fetch reports for a specific patient
Access: Doctor and Admin roles only

API Security

- Token-based authentication (JWT).
- Role-based access control for users.
- HTTPS for secure data transfer.

8. Authentication & Security

- Uses **JWT Authentication** for secure login sessions.
- Role-based access control:
    - **Admin:** Full system access.
    - **Doctor:** View and download reports.
    - **Technician:** Upload and classify images.
- Encryption ensures sensitive patient data protection.

---

9. User Interface

Modules Include:

- **Login / Register Page:** Secure authentication.
- **Image Upload Page:** File upload and preprocessing preview.
- **Dashboard:** Display of classification stats, reports, and analytics.
- **Reports Page:** Visual summary of blood cell counts and abnormalities.

UI is clean, responsive, and built using **React.js and CSS3**.

---

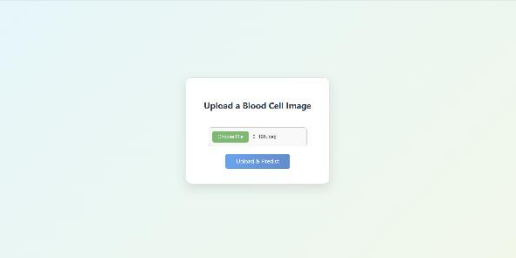
10. Testing

Testing performed includes:

- **Unit Testing:** Verified each component individually.
- **Functional Testing:** Checked login, upload, and classification workflows.
- **AI Model Testing:** Evaluated accuracy using medical datasets.

- **Security Testing:** Validated user roles and access restrictions.

11. Screenshots



---

12. Known Issues

- Very low-resolution images may reduce model accuracy.
- Internet dependency for MongoDB Atlas may cause delays.
- Mobile view optimization under progress.

---

13. Future Enhancements

- Integration with hospital management systems (HMS).
- Multi-language support for doctors and technicians.
- Enhanced analytics with disease prediction capabilities.
- AI model upgrade for new blood cell datasets.

14. Conclusion

HematoVision successfully automates **blood cell classification**, improves **diagnostic accuracy**, and reduces **manual errors** in laboratory workflows.
With AI-driven analysis and cloud-based storage, it represents a major step toward **modern digital pathology**.