Chapter 1

Individual Contribution

1.1 Problem Statement

Design and implement the backend API and face-recognition engine for the Attendance-Assistant system.

1.2 Student Details

Krishnaraj Thadesar
PRN: 1032210888
Roll Number: 15
Panel: A

1.3 Module Title

Backend & Face-Recognition Engine

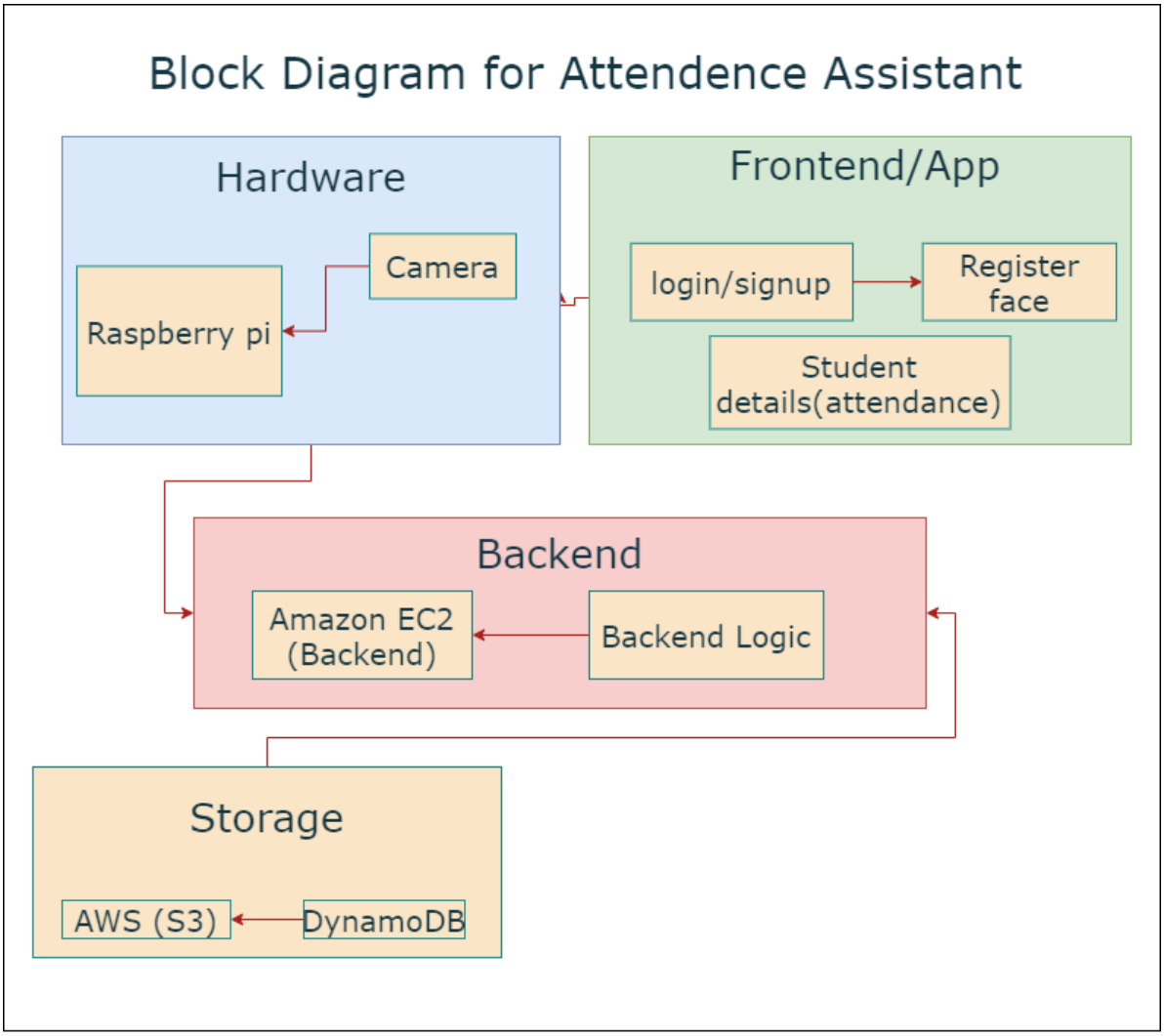

Figure 1.1: Block Diagram highlighting the backend & face-recognition module (Krishnaraj Thadesar's contribution).

1.4 Project Module Scope

End-to-end implementation of backend services, face-encoding storage and lookup, and concurrent API handling.

1.5 Project Modules – Individual Contribution

1. **Hardware & Software requirements:** PC (i7,16 GB RAM), server (4 vCPU,8 GB RAM); Python 3.9, FastAPI, face_recognition, Docker, MongoDB.
2. **Module Interfaces:**
   - POST `/api/auth/login`
   - POST `/api/faces/encode`
   - POST `/api/attendance/mark`
   - GET `/api/reports/daily`

3. **Module Dependencies:** face_recognition→dlib, numpy; FastAPI→uvicorn, pydantic; motor (MongoDB driver).
4. **Module Design:** Controller→Service→Model→Persistence layers; singleton model loader; JWT auth.
5. **Module Implementation:** Docker Compose, 1,200 LOC Python, integrated ResNet-based pipeline.
6. **Testing Strategies:** pytest (>=85% coverage), mocked CI tests, Postman smoke tests.
7. **Deployment:** Docker Compose (dev), AWS ECR/ECS Fargate (prod), auto-scaling.

Chapter 2

Individual Contribution

2.1 Problem Statement

Support the full-stack development cycle by contributing to UI design, API development, research, testing, and deployment for the Attendance-Assistant system.

2.2 Student Details

Parth Zarekar
PRN: 1032210846
Roll Number: 09
Panel: A

2.3 Module Title

Full-Stack Support & Research

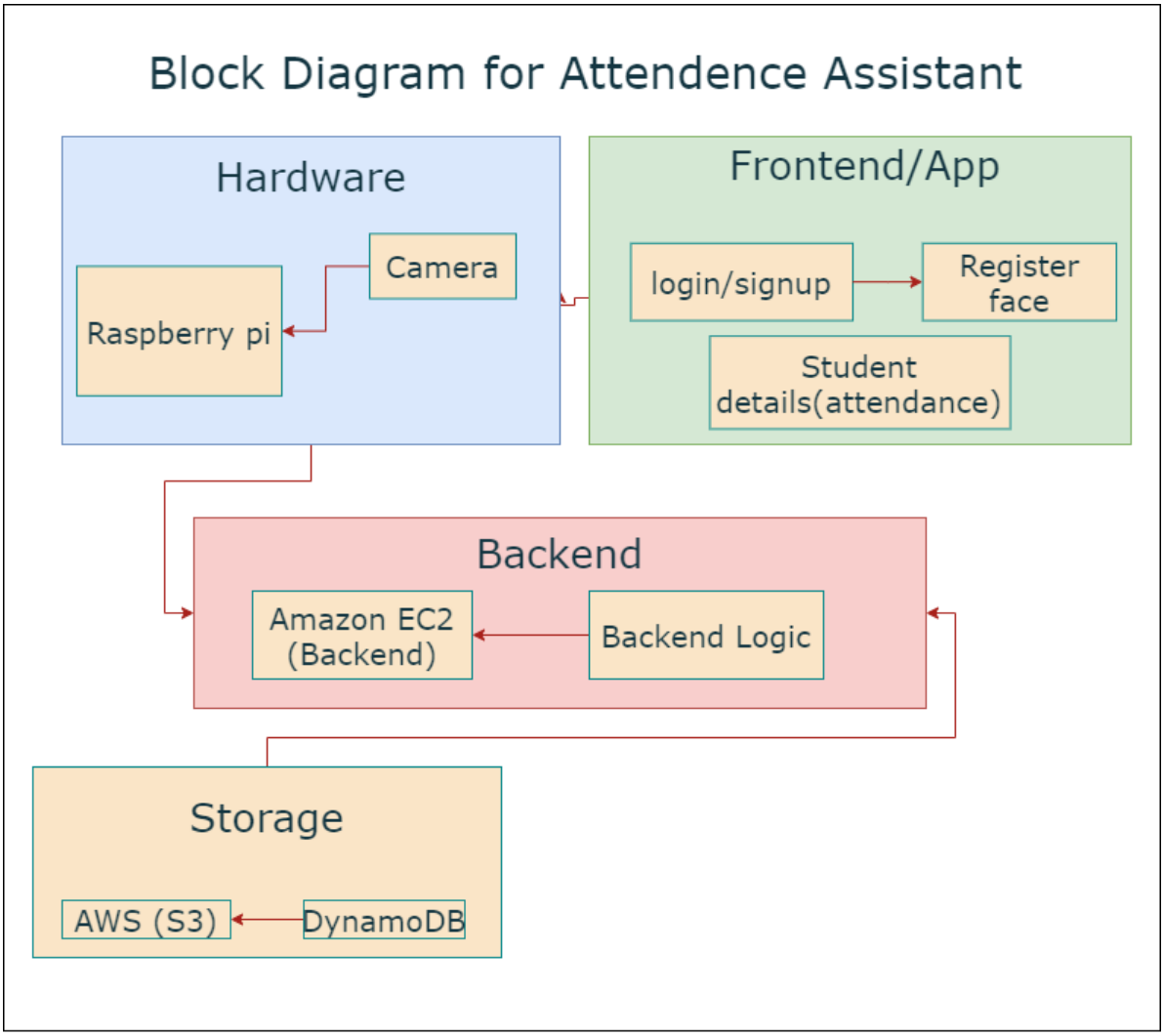

Figure 2.1: Block Diagram highlighting the modules supported by Parth Zarekar.

2.4 Project Module Scope

Assisted across UI design, backend API development, model-training research, paper drafting, testing, and deployment.

2.5 Project Modules – Individual Contribution

1. **Frontend:** Provided feedback and enhancements on Figma wireframes and UI flows.
2. **Backend API:** Implemented core endpoints for image upload, face encoding, and attendance marking.
3. **Model Research:** Supported training experiments and benchmark comparisons for face-recognition models.
4. **Literature Research:** Drafted and edited sections of the project research paper on algorithm selection.
5. **Testing:** Created and executed end-to-end tests (API smoke tests, basic UI checks).

6. **Deployment:** Deployed Dockerized services to a basic AWS environment and configured DynamoDB storage.

Chapter 3

Individual Contribution

3.1 Problem Statement

Evaluate and benchmark multiple face-recognition algorithms; support model selection and integration.

3.2 Student Details

Sourab Karad
PRN: 1032211150
Roll Number: 40
Panel: A

3.3 Module Title

Algorithm Research & Model Integration

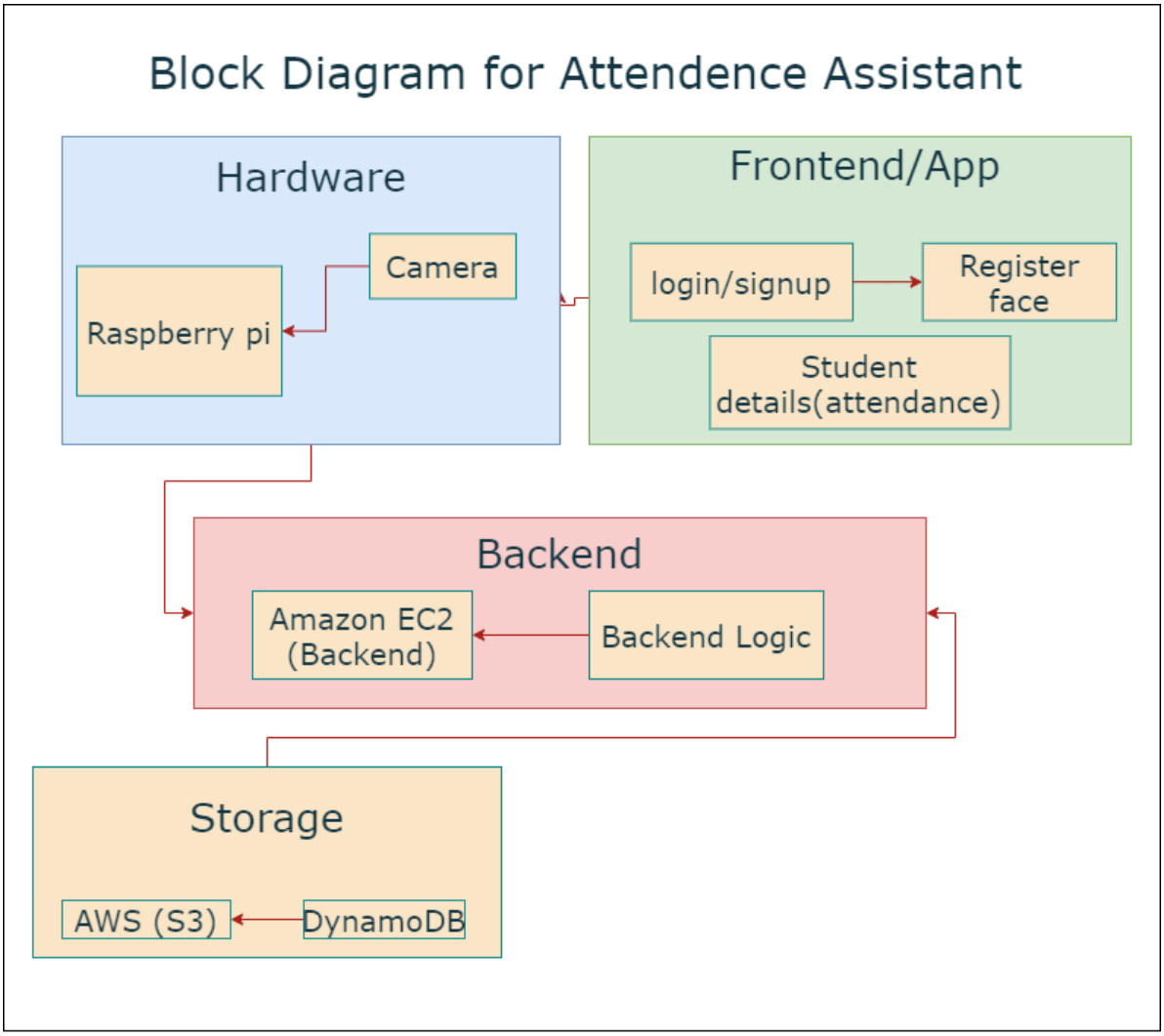

Figure 3.1: Block Diagram highlighting the algorithm research module (Sourab Karad's contribution).

3.4 Project Module Scope

Implementation and evaluation of face-recognition methods; performance reporting and API stub delivery.

3.5 Project Modules – Individual Contribution

1. **Hardware & Software requirements:** GPU (RTX 2060), dlib, OpenCV, torch, scikit-learn, pandas.
2. **Module Interfaces:** `train_model.py`, `evaluate.py`; JSON output (`accuracy, precision, recall`).
3. **Module Dependencies:** torch→torchvision; face_recognition→dlib; numpy→pandas.
4. **Module Design:** Abstract base classes; modular trainer & evaluator.
5. **Module Implementation:** 800 LOC benchmarking harness; comparative plots in report.
6. **Testing Strategies:** 5-fold cross-validation; confusion matrices.
7. **Deployment:** Packaged ResNet model as pickle; provided Dockerfile snippet.

Chapter 4

Individual Contribution

4.1 Problem Statement

Design and build the cross-platform mobile app for attendance marking via facial capture.

4.2 Student Details

Saubhagya Singh
PRN: 1032211144
Roll Number: 38
Panel: A

4.3 Module Title

Flutter Front-End Application

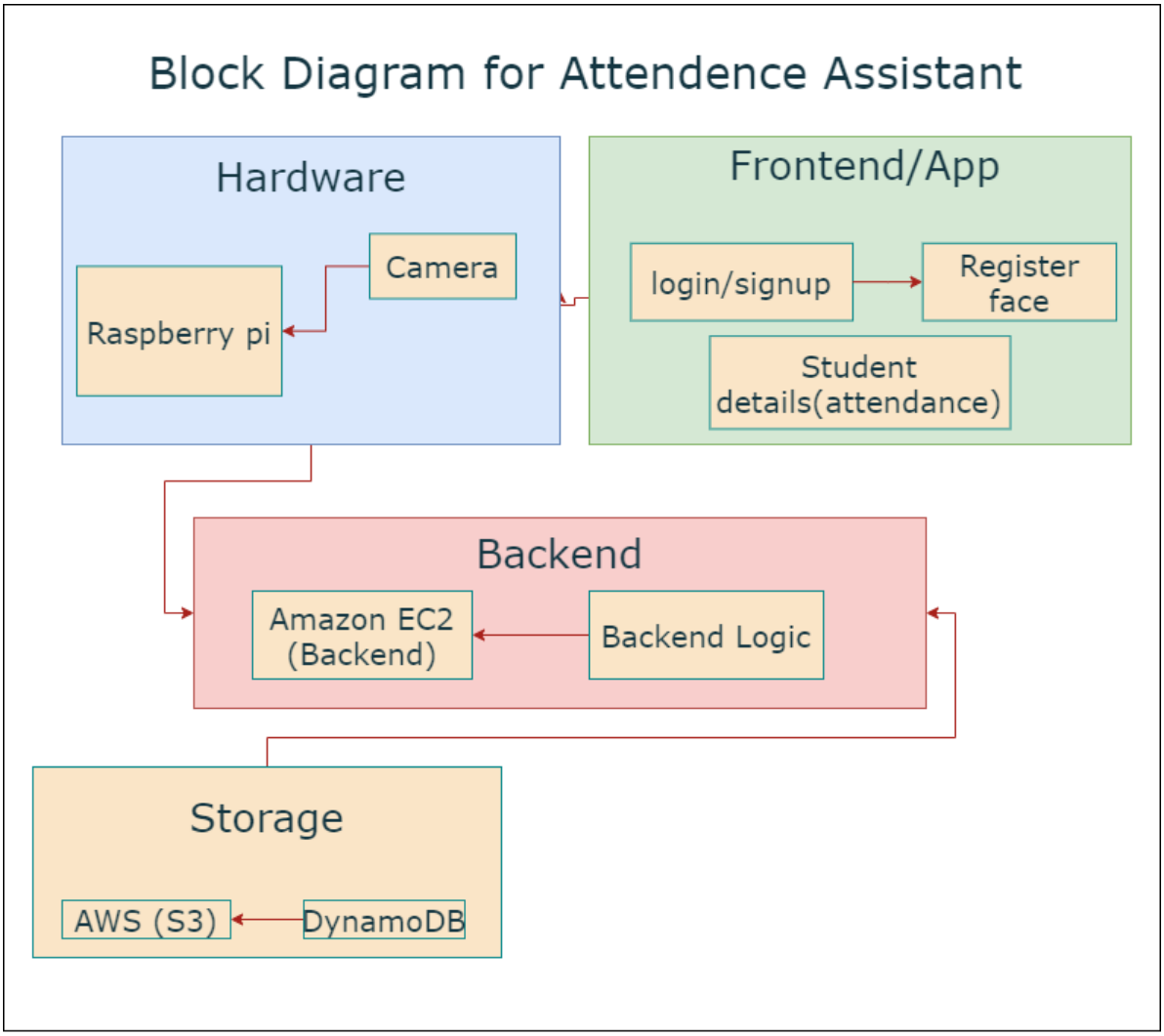

Figure 4.1: Block Diagram highlighting the frontend module (Saubhagya Singh's contribution).

4.4 Project Module Scope

Full-featured Flutter app: login, camera capture, attendance history, offline caching.

4.5 Project Modules – Individual Contribution

1. **Hardware & Software requirements:** Android/iOS device or emulator; Flutter 3.x, Dart, Android Studio, Xcode.
2. **Module Interfaces:** Flutter HTTP client→`POST /api/faces/encode`; Provider state management.
3. **Module Dependencies:** `camera`, `image_cropper`, `flutter_secure_storage`, `sqflite`.
4. **Module Design:** MVVM pattern; widget tree: Login→CameraView→AttendanceList.
5. **Module Implementation:** 1 500 LOC Dart; custom camera overlay; retry logic.
6. **Testing Strategies:** Widget tests; Flutter Driver integration tests.

7. **Deployment:** GH Actions→TestFlight & Play Store via Fastlane.