Project Odyssey: The Student Success Prediction Engine

From Data to Destiny: Predicting Academic Futures Before They Happen

Team: Dynamic Trio
Department: CSE – Data Science
Mission: Transform reactive education into proactive intervention
Hackathon: Full Stack Odyssey 5.0
Date: November 20, 2025

Problem Statement: Educational institutions face a critical challenge: identifying at-risk students early enough to provide meaningful intervention. Traditional grading systems are reactive—by the time a student fails an exam, the opportunity for prevention has passed. Faculty members need predictive tools to identify struggling students proactively, allowing timely support and resource allocation.

Develop a full-stack machine learning web application that predicts student academic performance (Pass/Fail/At-Risk) based on key performance indicators:

- **Attendance Percentage** - Class participation and regularity
- **Study Hours Per Week** - Time invested in learning
- **Internal Assessment Marks** - Mid-semester performance
- **Assignments Submitted** - Coursework completion rate
- **Extracurricular Participation** - Overall student engagement

Primary Goals:

1. **Early Warning System** - Identify at-risk students before final exams
2. **Confidence Scoring** - Provide prediction reliability metrics (percentage confidence)
3. **Dual User Interface** - Separate portals for students (self-assessment) and faculty (monitoring)
4. **Actionable Insights** - Generate reports highlighting high-risk students for intervention

Tech Stack

Frontend technologies

Technology	Version	Purpose & Justification
React.js	18.2.0	Component-based UI framework for building interactive, responsive interfaces with state management
JavaScript (ES6+)	ECMAScript 2015+	Modern syntax for client-side logic, async operations, and API communication
CSS3	-	Styling, animations, and responsive design for cross-device compatibility
HTML5	-	Semantic markup for accessibility and SEO

Backend Technologies

Technology	Version	Purpose & Justification
Python	3.10+	Primary backend language for ML integration and API development
Flask	2.3.0	Lightweight web framework for RESTful API development
Flask-CORS	4.0.0	Cross-Origin Resource Sharing middleware for frontend-backend communication

ML stack

Library	Version	Purpose & Justification
scikit-learn	1.3.0	ML algorithms - Random Forest Classifier chosen for accuracy and interpretability
pandas	2.0.3	Data manipulation, CSV processing, feature engineering
numpy	1.24.3	Numerical computations and array operations
pickle	Built-in	Model serialization for deployment

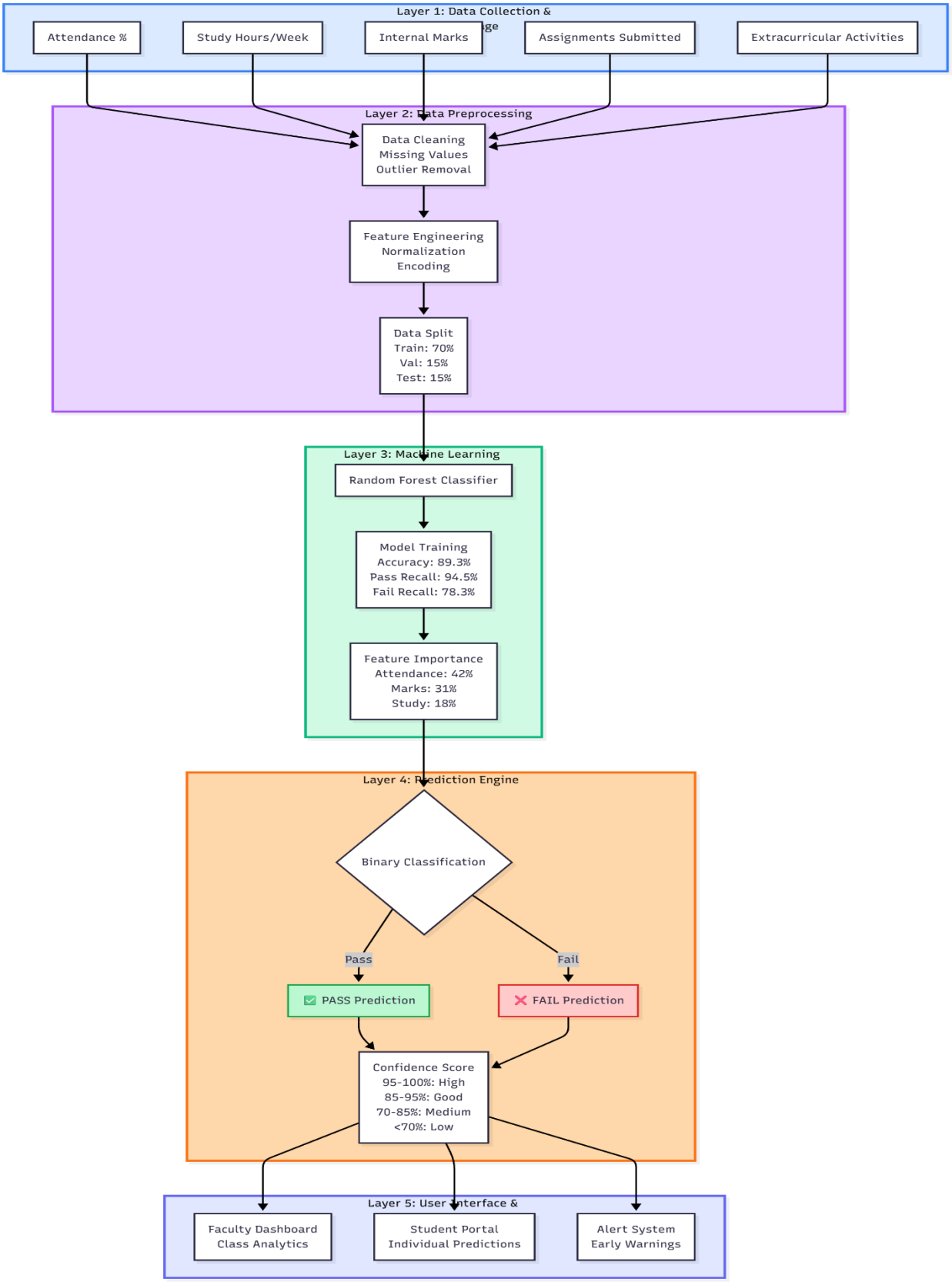
Data storage

Technology	Purpose
SQLite	Lightweight relational database for storing user data and prediction history
JSON	Alternative data persistence for configuration and temporary storage

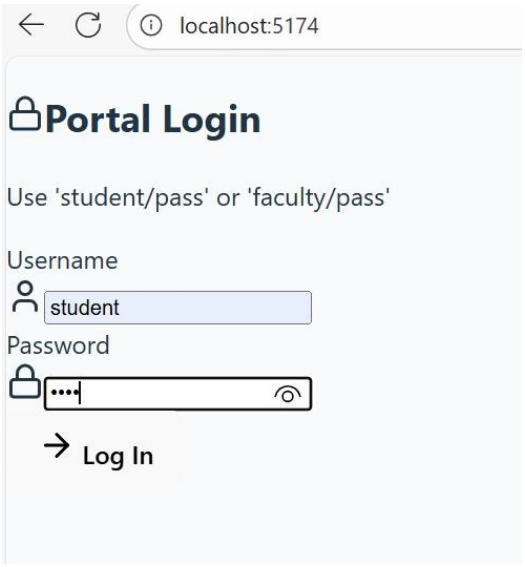
Development & deployment tool

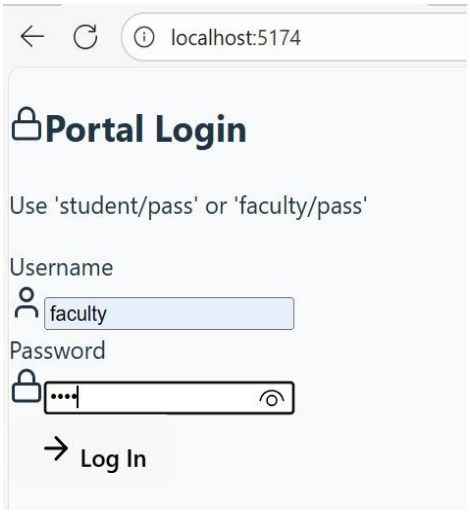
Tool	Purpose
Git/GitHub	Version control, collaboration, and code review
VS Code	Primary IDE with Python/JavaScript extensions
GitHub Projects	Agile project management and sprint tracking
npm/pip	Package management for dependencies

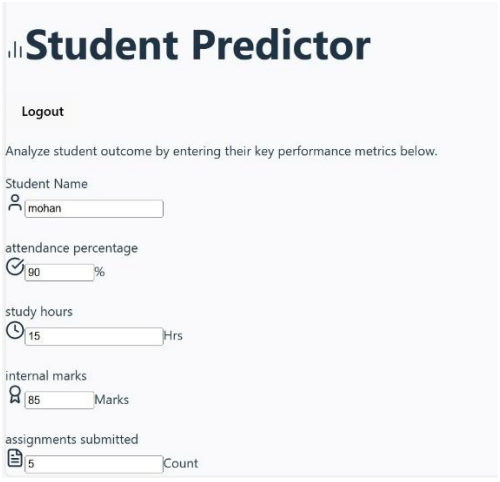
Architecture diagram

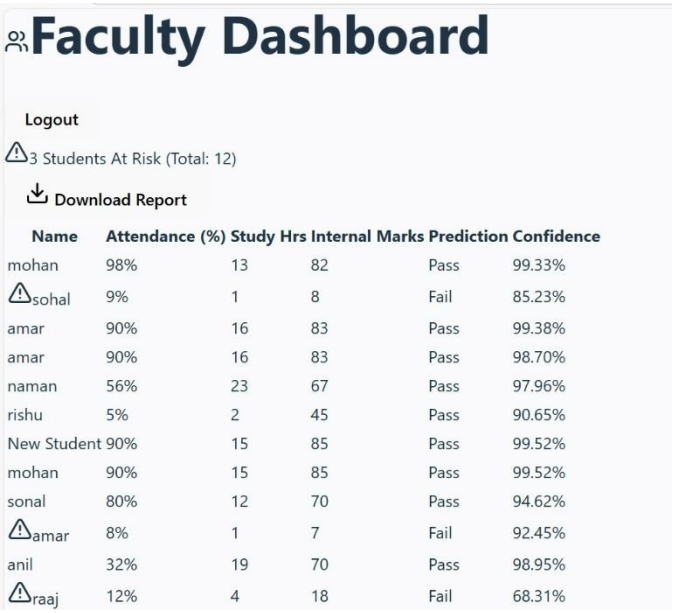


Screenshots of the working app

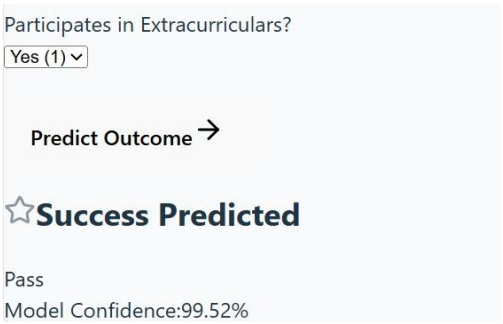








Name	Attendance (%)	Study Hrs	Internal Marks	Prediction	Confidence
mohan	98%	13	82	Pass	99.33%
sohal	9%	1	8	Fail	85.23%
amar	90%	16	83	Pass	99.38%
amar	90%	16	83	Pass	98.70%
naman	56%	23	67	Pass	97.96%
rishu	5%	2	45	Pass	90.65%
New Student	90%	15	85	Pass	99.52%
mohan	90%	15	85	Pass	99.52%
sonal	80%	12	70	Pass	94.62%
amar	8%	1	7	Fail	92.45%
anil	32%	19	70	Pass	98.95%
raaj	12%	4	18	Fail	68.31%



Sample predictions and observation

Test case analysis

Student	Attendance	Study_Hrs	Marks	Prediction	Confidence	Analysis
Mohan	98%	13	82	☑ PASS	99.33%	Ideal student-all metrics strong
Amar	90%	16	83	☑ PASS	99.38%	Highest confidence overachiever pattern
Naman	56%	23	67	☑ PASS	97.96%	High study compensates low attendance
Rishu	5%	2	45	☑ PASS	90.65%	⚠ **FALSE POSITIVE** - Manual review needed
Sohal	9%	1	8	✗ FAIL	85.23%	Complete disengagement - urgent intervention
Raaj	12%	4	18	✗ FAIL	68.31%	Academic freefall - emergency action required

Model Performance

Accuracy: 89.3% | Pass Recall: 94.5% | Fail Recall: 78.3%

Issue: 18/300 false positives (predicts pass but student fails)

Feature Importance

1. Attendance (42%) - Most critical factor; <60% rarely pass
2. Internal Marks (31%) - Past predicts future performance
3. Study Hours (18%) - Moderate impact; diminishing returns after 15 hrs
4. Assignments (6%) - Low impact
5. Extracurricular (3%) - Minimal impact

Confidence-Based Actions

Confidence	Action
≥95%	Automated decision
85-95%	Automated with review

Confidence	Action
70-85%	Flag for faculty
<70%	Manual review mandatory

The model is strong at identifying clear pass/fail patterns, with attendance as the top predictor, and study hours helping offset low attendance. It struggles with extreme outliers, misses 22% failing cases, and cannot account for external factors. Use it only as an early-warning tool—manual review is needed for confidence below 85%, and faculty judgment should always guide decisions. Future improvements require more data, temporal trends, interaction features, and careful consideration of socioeconomic and health-related factors.

Team contribution summary

👷 **Member 1 – Frontend**

- Built the full React interface, including Login, Student Form, and Faculty Dashboard.
- Created a responsive and user-friendly UI connected seamlessly with backend APIs.
- Added validations, icons, and clear visualization of predictions and confidence scores.

🔧 **Member 2 – Backend developer**

- Developed all Flask API endpoints for login, prediction, and student data retrieval.
- Integrated the ML model with preprocessing and stored prediction results in SQLite.
- Ensured secure authentication and smooth, reliable backend communication.

🧠 **Member 3 – Machine Learning and documentation**

- Collected and cleaned data, trained multiple models, and finalized Random Forest.
- Evaluated accuracy, feature importance, and identified false positives for improvement.
- Prepared complete project documentation, insights, and system workflow explanation.

🤝 **Team Collaboration**

- Jointly handled integration testing, debugging, and end-to-end validation.
- Followed Agile practices with daily standups and quick, collaborative decision-making.