Studistic

تحت اشراف

أ. د \ خالد فتحي

(IS) **عمرو خالد منصور حسن** – نظم معلومات

(IS) **شيماء عيد موسى** – نظم معلومات

(SE) **حسام حسين عبد الستار** – هندسة برمجيات

(IS) **سهام سيد شعيب** – نظم معلومات

كلية الحاسبات والمعلومات - جامعة اسيوط الاهليه

An AI-Powered Student Performance Monitoring & Early-Warning Web System........... 2
1. Project Overview........ 2
1.1 Problem Statement........ 2
1.2 Why Student Performance Prediction Matters........ 3
2. System Architecture........ 3
2.1 Architectural Overview........ 3
2.2 High-Level Architecture Diagram........ 4
2.3 Component Description........ 4
User........ 4
Web Application (Frontend)........ 4
Backend API........ 5
Machine Learning Module........ 5
Supabase........ 5
2.4 Data Flow Diagram (DFD)........ 6
3. Database Design........ 6
3.1 Overview........ 6
3.2 Entity Relationship Diagram (ERD)........ 7
4. Machine Learning Module........ 7
4.1 Models Used........ 7
4.2 Why Linear Regression Was Chosen........ 8
4.3 Risk Classification Logic........ 8
4.4 SHAP Explainability........ 8
4.5 ML Integration Workflow........ 9
5. Web Application Features........ 11
Dashboard........ 11
Profile........ 11
To-Study (Kanban)........ 11
Grades........ 11
Percentages........ 11
Comparisons........ 12
Logout........ 12
6. Security & Privacy Considerations........ 12
7. Technology Stack........ 12
8. Future Enhancements........ 12
9. Conclusion........ 13

Studistic

An AI-Powered Student Performance Monitoring & Early-Warning Web System

> Turning student data into early insights, actionable guidance, and academic success.

---

1. Project Overview

1.1 Problem Statement

Academic underperformance remains a persistent challenge across educational institutions. Students often struggle due to a combination of academic, behavioral, lifestyle, and socioeconomic factors such as low attendance, limited study hours, lack of motivation, inadequate learning resources, poor sleep patterns, or insufficient parental involvement.

In most cases, these issues are only discovered **after final exam results**, when it becomes difficult to provide meaningful intervention. Traditional evaluation methods depend mainly on final grades and manual observation, which are **reactive**, time-consuming, and unable to uncover hidden patterns within student data.

There is a clear need for an **intelligent, proactive, and data-driven system** that can:

- Predict student performance in advance
- Detect academic risk early
- Provide explainable insights and personalized recommendations

Studistic addresses this gap by combining **machine learning** with a **web-based platform** to support early academic intervention.

1.2 Why Student Performance Prediction Matters

Predicting student performance enables institutions and students to shift from **reaction to prevention**.

By analyzing factors such as study habits, attendance, previous grades, lifestyle balance, and learning environment, machine learning models can identify patterns that are not easily observable through traditional analysis.

Early prediction enables:

- **Early identification of at-risk students** before failure occurs
- **Personalized recommendations** based on individual weaknesses
- **Data-driven decisions** for educators and institutions
- **Improved student engagement** through transparent feedback
- **Efficient resource allocation** for tutoring and academic support

Integrating these predictions into a web application transforms raw data into a **practical decision-support system** accessible anytime, anywhere.

2. System Architecture

2.1 Architectural Overview

Studistic follows a **modular, scalable architecture** that cleanly separates:

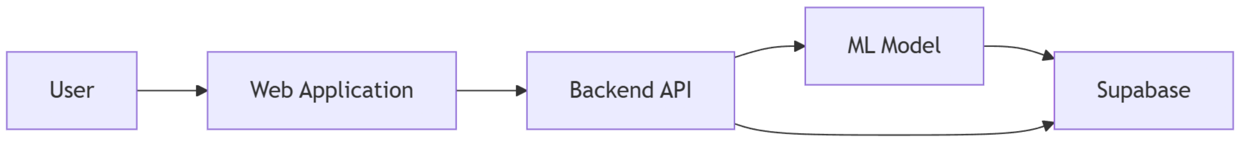
- Presentation layer
- Business logic
- Machine learning processing
- Data storage and authentication

This improves **security, maintainability, and extensibility**.

Core Components:

1. User
2. Web Application (Frontend)
3. Backend API
4. Machine Learning Module
5. Supabase (Database, Auth, Storage)

2.2 High-Level Architecture Diagram



2.3 Component Description

User

The student who interacts with the system to:

- View academic performance
- Receive predictions and risk levels
- Track progress and recommendations

Web Application (Frontend)

Responsibilities:

- Sign up / Sign in
- Dashboard and visualizations
- Predictions and recommendations
- Sidebar navigation (Profile, Grades, Tasks, Comparisons)

The frontend communicates **only with the backend API**, ensuring security and abstraction.

Backend API

Acts as the system's control layer:

- Handles authenticated requests
- Retrieves and updates student data
- Sends features to the ML model
- Stores predictions and insights

Machine Learning Module

Responsible for:

- Predicting exam scores
- Classifying academic risk
- Generating explainable insights

The model is **server-side only**, protecting integrity and preventing misuse.

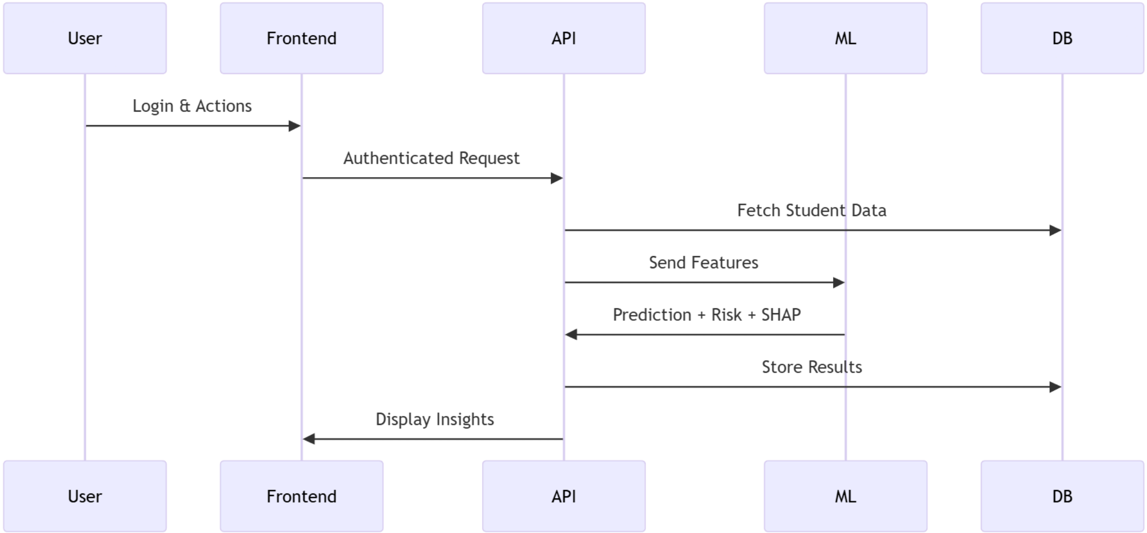
Supabase

Provides:

- **Authentication** (secure login & sessions)
- **Database** (students, features, grades, predictions)
- **Storage** (profile images)

Supabase enables rapid development with production-ready scalability.

2.4 Data Flow Diagram (DFD)



---

3. Database Design

3.1 Overview

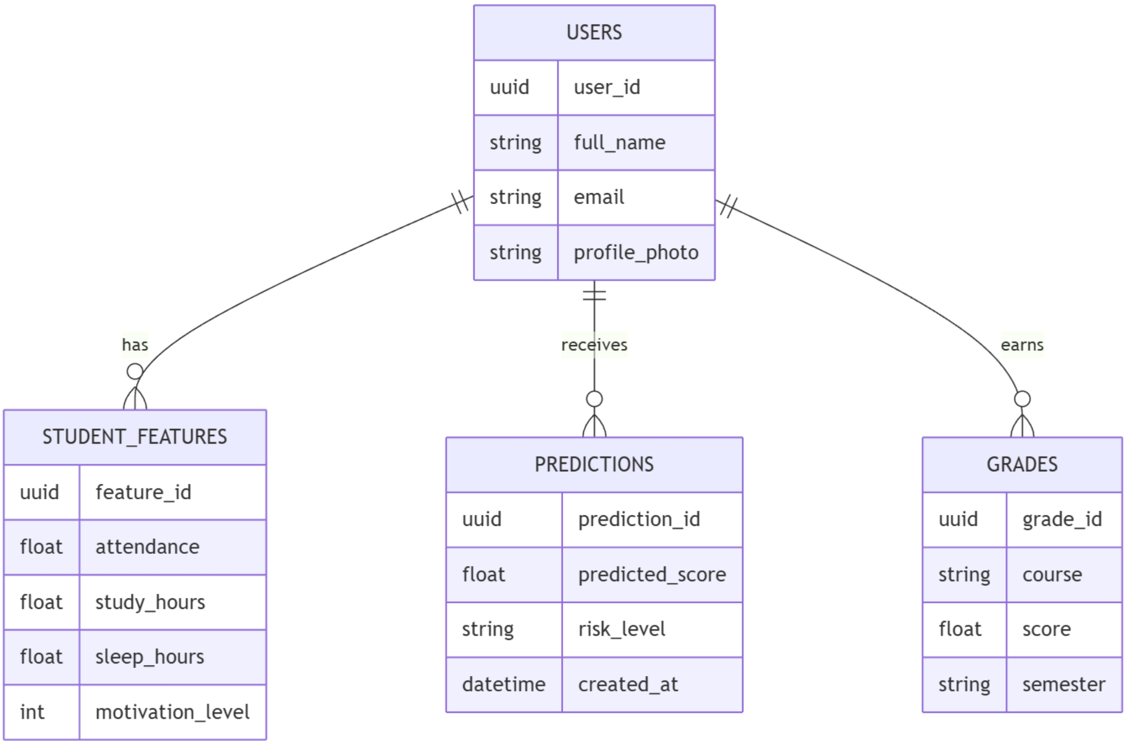
The database is designed to:

- Support secure authentication
- Store ML inputs and outputs
- Enable performance tracking over time

Main Tables:

- users
- student_features
- predictions
- grades

3.2 Entity Relationship Diagram (ERD)



USERS	
uuid	user_id
string	full_name
string	email
string	profile_photo

STUDENT_FEATURES	
uuid	feature_id
float	attendance
float	study_hours
float	sleep_hours
int	motivation_level

PREDICTIONS	
uuid	prediction_id
float	predicted_score
string	risk_level
datetime	created_at

GRADES	
uuid	grade_id
string	course
float	score
string	semester

---

4. Machine Learning Module

4.1 Models Used

- **Linear Regression**
- Random Forest Regressor
- Artificial Neural Network (ANN)

Models were evaluated using:

- R^2
- RMSE
- MAE

4.2 Why Linear Regression Was Chosen

Linear Regression was selected because it offers:

- Strong predictive performance
- Low overfitting risk
- High interpretability
- Seamless SHAP integration

In an educational system, **explainability is as important as accuracy**.

4.3 Risk Classification Logic

Risk Level	Predicted Score
High Performer	≥ 75
Medium Risk	60 – 74
At Risk	< 60

This transforms raw predictions into **actionable insights**.

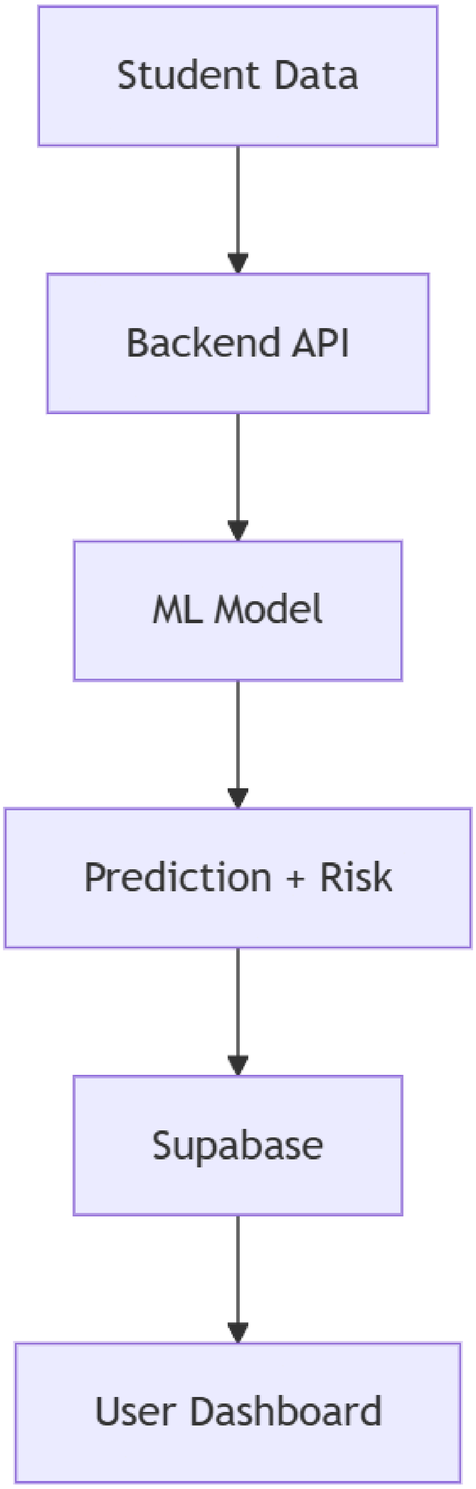
4.4 SHAP Explainability

SHAP enables:

- **Global explanations** (most important features overall)
- **Local explanations** (why a student got this score)

This builds **trust, transparency, and actionable guidance**.

4.5 ML Integration Workflow



---

5. Web Application Features

Dashboard

Shows: prediction, risk level, recommendations
Why: instant academic awareness

Profile

Shows: name, photo, basic info
Why: personalization & identity

To-Study (Kanban)

Shows: tasks in To-Do / In-Progress / Done
Why: improves study discipline

Grades

Shows: all course grades
Why: track academic history

Percentages

Shows: feature contributions
Why: understand performance drivers

Comparisons

Shows: performance vs peers (anonymized)
Why: context and motivation

Logout

Why: security and session control

6. Security & Privacy Considerations

- JWT-based authentication (Supabase)
- Role-based access control
- No direct ML access from frontend
- Aggregated peer comparisons only
- Encrypted data storage

7. Technology Stack

Layer	**Technology**
Frontend	React / Next.js
Backend	Node.js / Python API
ML	Scikit-learn, SHAP
Database	Supabase (PostgreSQL)
Auth	Supabase Auth

8. Future Enhancements

- Educator dashboard
- Real-time alerts
- Mobile app version
- More advanced ML models
- Adaptive recommendations

---

9. Conclusion

Studistic transforms student data into **early warnings, insights, and guidance**.
By combining machine learning, explainability, and a modern web application, the system empowers students to take control of their academic journey **before failure occurs**.

Studistic — Predict Early. Act Smarter. Succeed Faster.