Big-Data
Edu-Predict
Documentation

Faculty:
Sir Aseef Ahmed

Presented for:
E-Project

Our Team

Mustufa Kashif

Saad Sarfaraz

S.M. Hamza

Muneeb Khan

Owais Hayat

Zaryab Khan

Rehman Ahmed

Mentor & Guide

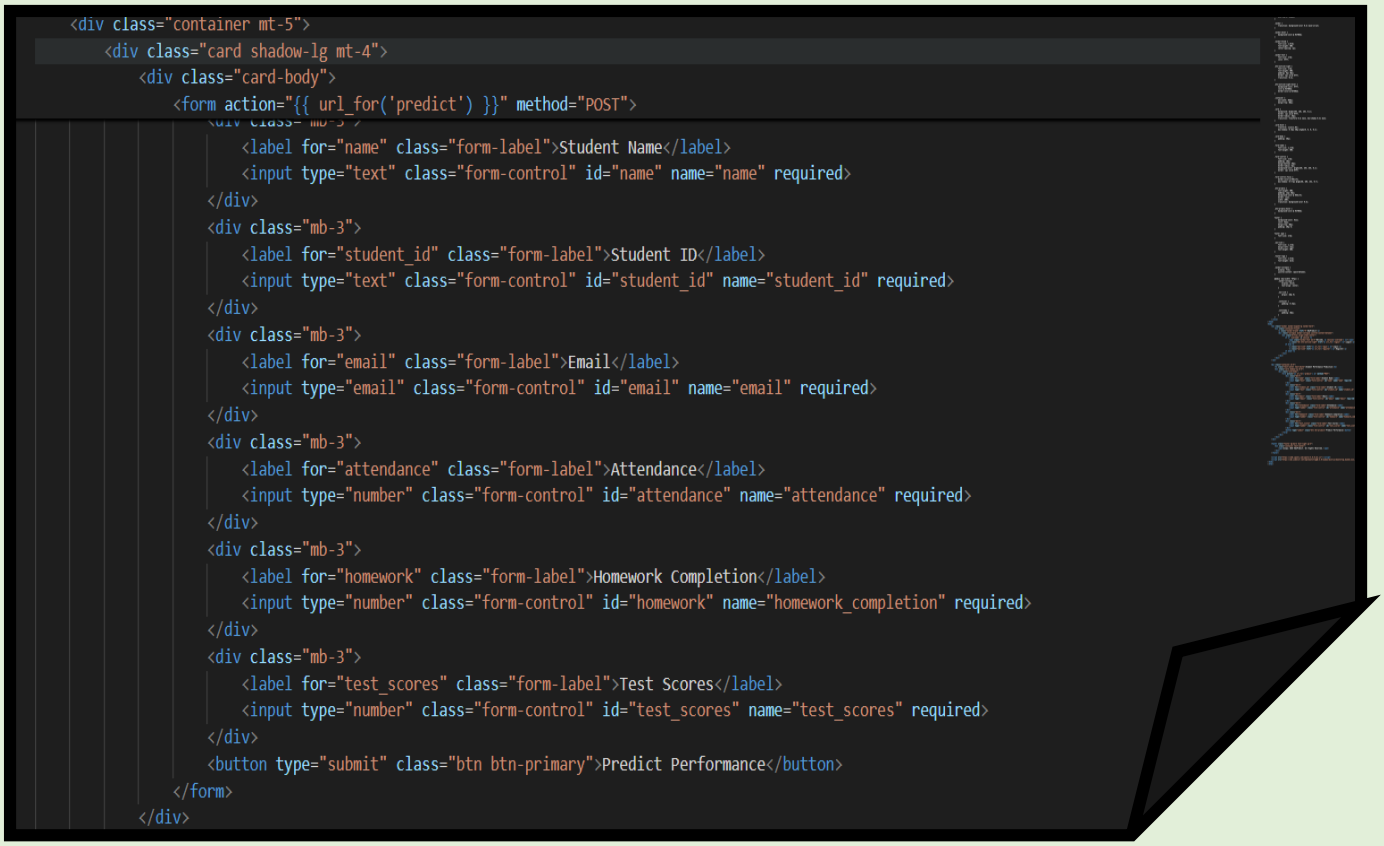
Sir Aseef Ahmed

TABLE OF CONTENTS

- Introduction
- System Architecture
- Authentication & Authorization
- Data Ingestion
- ETL Workflow
- Data Storage
- Data Processing
- Machine Learning Models
- Data Visualization
- Notification & Alerting
- Feedback and Support
- Non-Functional Implementation
- DevOps & Deployment
- Documentation & Training
- Appendices

1. Introduction:

- **Purpose of the Guide**:
  - Assist developers in understanding, building, deploying, and maintaining EduPredict.
- **Scope**:
  - Covers backend, frontend, data processing, machine learning, and deployment aspects.
- **Audience**:
  - Data engineers, software developers, ML engineers, and DevOps teams.



```
<div class="container mt-5">
    <div class="card shadow-lg mt-4">
        <div class="card-body">
            <form action="{{ url_for('predict') }}" method="POST">
                <div class="mb-3">
                    <label for="name" class="form-label">Student Name</label>
                    <input type="text" class="form-control" id="name" name="name" required>
                </div>
                <div class="mb-3">
                    <label for="student_id" class="form-label">Student ID</label>
                    <input type="text" class="form-control" id="student_id" name="student_id" required>
                </div>
                <div class="mb-3">
                    <label for="email" class="form-label">Email</label>
                    <input type="email" class="form-control" id="email" name="email" required>
                </div>
                <div class="mb-3">
                    <label for="attendance" class="form-label">Attendance</label>
                    <input type="number" class="form-control" id="attendance" name="attendance" required>
                </div>
                <div class="mb-3">
                    <label for="homework" class="form-label">Homework Completion</label>
                    <input type="number" class="form-control" id="homework" name="homework_completion" required>
                </div>
                <div class="mb-3">
                    <label for="test_scores" class="form-label">Test Scores</label>
                    <input type="number" class="form-control" id="test_scores" name="test_scores" required>
                </div>
                <button type="submit" class="btn btn-primary">Predict Performance</button>
            </form>
        </div>
```

System Architecture Overview

- High-level Diagram: Show interactions between components (HDFS, Kafka, Spark, ML, UI, etc.).
- Components:
  - Authentication & Authorization
  - Data Ingestion (Batch & Real-time)
  - Storage (HDFS)
  - Data Processing (Apache Spark/Hive)
  - Machine Learning (Python/MLlib/Scikit-learn)
  - Visualization (Grafana/Tableau/custom dashboards)
  - Notification Service
  - Feedback & Support Module

```
<div class="container mt-5">
    <h1 class="text-center text-white mb-4">Login to EduPredict</h1>
    <div class="row justify-content-center">
        <div class="col-md-6">
            <div class="card">
                <div class="card-body">
                    <form action="{{ url_for('login') }}" method="POST">
                        <div class="mb-3">
                            <label for="username" class="form-label text-white">Username</label>
                            <input type="text" class="form-control" id="username" name="username" required>
                        </div>
                        <div class="mb-3">
                            <label for="password" class="form-label text-white">Password</label>
                            <input type="password" class="form-control" id="password" name="password" required>
                        </div>
                        <button type="submit" class="btn btn-primary w-100">Login</button>
                        <div class="mt-3 text-center">
                            <a href="{{ url_for('register') }}" class="text-white">Don't have an account? Register here</a>
                        </div>
                    </form>
                </div>
            </div>
        </div>
    </div>
</div>
```

3. Authentication & Authorization

- Tech Stack: OAuth 2.0 / JWT / Spring Security / Firebase Auth
- Roles: Admin, Teacher, Student, Analyst
- RBAC Implementation: Secure endpoints, role-based access checks

```
<div class="container mt-5">
    <h1 class="text-center mb-4">Create an EduPredict Account</h1>
    <div class="card shadow-sm mt-4">
        <div class="card-body">
            <form action="{{ url_for('register') }}" method="POST">
                <div class="mb-3">
                    <label for="username" class="form-label">Username</label>
                    <input type="text" class="form-control" id="username" name="username" required>
                </div>
                <div class="mb-3">
                    <label for="email" class="form-label">Email address</label>
                    <input type="email" class="form-control" id="email" name="email" required>
                </div>
                <div class="mb-3">
                    <label for="password" class="form-label">Password</label>
                    <input type="password" class="form-control" id="password" name="password" required>
                </div>
                <button type="submit" class="btn btn-primary w-100">Register</button>
                <div class="mt-3 text-center">
                    <a href="{{ url_for('login') }}" class="text-decoration-none">Already have an account? Login here</a>
                </div>
            </form>
        </div>
    </div>
</div>
```

4. Data Ingestion:

- Overview of existing educational analytics platforms.
- Tools used in academic performance prediction.
- Use of big data frameworks (e.g., Hadoop, Spark, Kafka) in similar domains.
- Comparative analysis with EduPredict in terms of innovation and scalability.

5. Data Storage:

- HDFS Directory Structure:
  - /raw_data/
  - /processed_data/
  - /ml_models/
- Partitioning Strategy: By date, student ID, institution
- Backup Strategy: Hadoop Snapshots, scheduled HDFS backup

6. Data Processing:

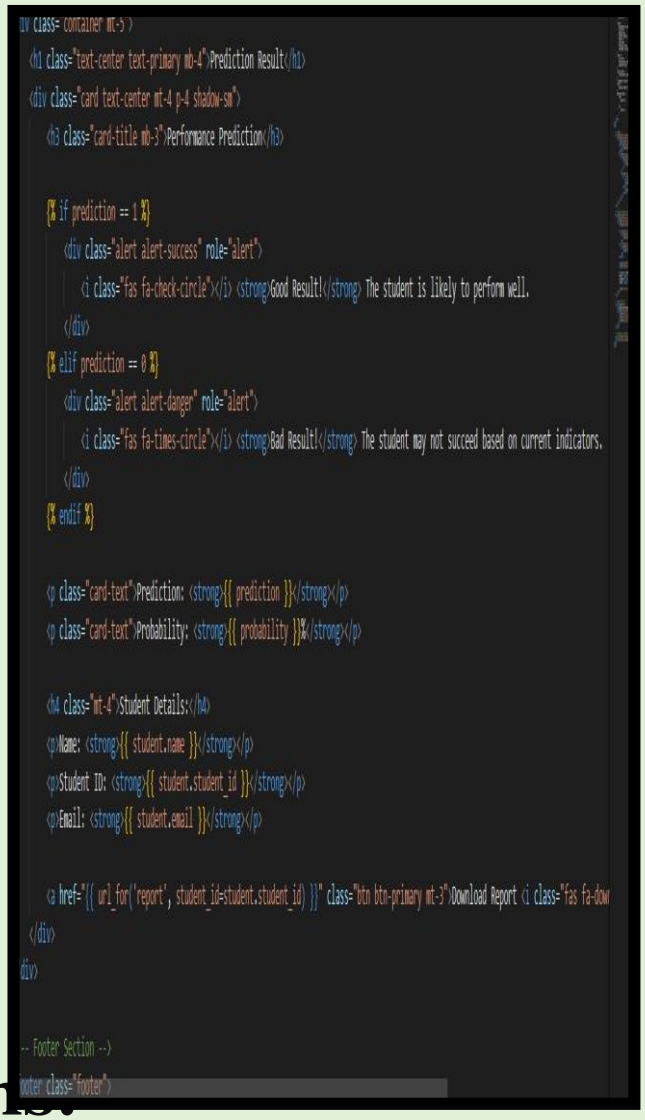
- **Batch Processing: Apache Spark jobs (PySpark or Scala)**
- **Real-time Processing: Apache Spark Streaming or Kafka Streams**
- **Data Cleaning & Handling Missing Values: Use Spark's DataFrame APIs**
- **Anomaly Detection Rules: Statistical thresholds, z-score, or ML-based**

7. Machine Learning Models:

- **Use Cases:**
  a. **Predicting Dropout Rates**
  b. **Student Performance Trends**
  c. **Course Demand Forecasting**
- **Tech Stack:**
  a. **Python, Scikit-learn, MLlib**
- **Model Lifecycle:**
  a. **Data Preparation**
  b. **Feature Engineering**
  c. **Training & Validation**
  d. **Model Deployment (Flask/MLFlow)**
  e. **Model Retraining Strategy**

8. Data Visualization:

- **Tools:**
  - **i. Grafana, Power BI**
  - **ii. Tableau, or React.js + Chart.js**
- **Dashboard Features:**
  - **i. Role-based views**
- **Filters:**
  - **i. time range, student category, academic year**
- **Export options (PDF, Excel)**



```
<div class="container mt-5">
  <h1 class="text-center text-primary mb-4">Prediction Result</h1>
  <div class="card text-center mt-4 p-4 shadow-sm">
    <h3 class="card-title mb-3">Performance Prediction</h3>

    {% if prediction == 1 %}
      <div class="alert alert-success" role="alert">
        <i class="fas fa-check-circle"></i> <strong>Good Result!</strong> The student is likely to perform well.
      </div>
    {% elif prediction == 0 %}
      <div class="alert alert-danger" role="alert">
        <i class="fas fa-times-circle"></i> <strong>Bad Result!</strong> The student may not succeed based on current indicators.
      </div>
    {% endif %}

    <p class="card-text">Prediction: <strong>{{ prediction }}</strong></p>
    <p class="card-text">Probability: <strong>{{ probability }}%</strong></p>

    <h4 class="mt-4">Student Details:</h4>
    <p>Name: <strong>{{ student.name }}</strong></p>
    <p>Student ID: <strong>{{ student.student_id }}</strong></p>
    <p>Email: <strong>{{ student.email }}</strong></p>

    <a href="{{ url_for('report', student_id=student.student_id) }}" class="btn btn-primary mt-3">Download Report <i class="fas fa-dow
  </div>
</div>


<!-- Footer Section -->
<footer class="footer">
```

9. Notification & Alerting:

- **Mechanism: Apache Airflow or custom job triggers**
- **Notification Channels: Email, SMS, Webhooks**
- **Alert Types: Attendance below threshold, grade drops, inactivity**

10. Challenges & Limitations:

- **Modules:**
- **In-app chat (using tools like Intercom or custom)**
- **Ticketing system (integrate with Jira/ServiceNow or build internal)**
- **Feedback Forms stored in DB for analysis**

11. Non-Functional Implementation:

- **Performance:**
  - **Spark job optimization (partitioning, caching)**
  - **HDFS tuning**
- **Security:**
  - **Data encryption: TLS for transmission, AES for storage**
  - **Masking and anonymization of student data**
- **Reliability:**
  - **Failover handling with Hadoop YARN**
  - **Scheduled backups**
- **Scalability:**
  - **Horizontal scaling of Kafka/Spark**
  - **Load balancers in web/API layers**
- **Monitoring:**
  - **Prometheus + Grafana**
  - **Hadoop/Spark resource managers**

12. DevOps & Deployment:

- **CI/CD Tools: Jenkins, GitHub Actions**
- **Containerization: Docker for services**
- **Orchestration: Kubernetes (optional)**
- **Environments: Dev, Staging, Production**

```
<h1 class="text-center text-primary mb-4">Prediction Result</h1>
<div class="card text-center mt-4 p-4 shadow-sm">
    <h3 class="card-title mb-3">Performance Prediction</h3>

    {% if prediction == 1 %}
        <div class="alert alert-success" role="alert">
            <i class="fas fa-check-circle"></i> <strong>Good Result!</strong> The student is likely to perform well.
        </div>
    {% elif prediction == 0 %}
        <div class="alert alert-danger" role="alert">
            <i class="fas fa-times-circle"></i> <strong>Bad Result!</strong> The student may not succeed based on current indicators.
        </div>
    {% endif %}

    <p class="card-text">Prediction: <strong>{{ prediction }}</strong></p>
    <p class="card-text">Probability: <strong>{{ probability }}%</strong></p>

    <h4 class="mt-4">Student Details:</h4>
    <p>Name: <strong>{{ student.name }}</strong></p>
    <p>Student ID: <strong>{{ student.student_id }}</strong></p>
    <p>Email: <strong>{{ student.email }}</strong></p>

    <a href="{{ url_for('report', student_id=student.student_id) }}" class="btn btn-primary mt-3">Download Report <i class="fas fa-dow
</div>

<!-- Footer Section -->
<footer class="footer">
```

13. Documentation & Training Materials:

- **Developer Docs: Hosted on GitHub Pages, Docusaurus or ReadTheDocs**
- **User Docs: Tutorials, FAQs, role-based guides**
- **Training Video: Record via OBS Studio with voiceover, walkthrough of each module**

14. Appendices:

- **API Reference**
- **Data Dictionary**
- **Sample Config Files**
- **Sample Datasets**

```
div class= container mt-5 >
  <h1 class="text-center text-primary mb-4">Prediction Result</h1>
  <div class="card text-center mt-4 p-4 shadow-sm">
      <h3 class="card-title mb-3">Performance Prediction</h3>

      {% if prediction == 1 %}
          <div class="alert alert-success" role="alert">
              <i class="fas fa-check-circle"></i> <strong>Good Result!</strong> The student is likely to perform well.
          </div>
      {% elif prediction == 0 %}
          <div class="alert alert-danger" role="alert">
              <i class="fas fa-times-circle"></i> <strong>Bad Result!</strong> The student may not succeed based on current indicators.
          </div>
      {% endif %}

      <p class="card-text">Prediction: <strong>{{ prediction }}</strong></p>
      <p class="card-text">Probability: <strong>{{ probability }}%</strong></p>

      <h4 class="mt-4">Student Details:</h4>
      <p>Name: <strong>{{ student.name }}</strong></p>
      <p>Student ID: <strong>{{ student.student_id }}</strong></p>
      <p>Email: <strong>{{ student.email }}</strong></p>

      <a href="{{ url_for('report', student_id=student.student_id) }}" class="btn btn-primary mt-3">Download Report <i class="fas fa-dowr
  </div>
/div>

!-- Footer Section -->
footer class="footer">
```

THANKS!