**Comprehensive Development Plan for Autogradex (Single-Phase)**

This plan outlines the development of Autogradex, a comprehensive evaluation and grading system, in a single, iterative phase.

**Phase 1: Core Development and Initial Launch**

**1. User Management and Role-Based Access Control**

* **Backend:**
  + Develop robust user registration and login functionalities.
  + Implement secure authentication and authorization mechanisms.
  + Create a comprehensive database schema for users, roles (Admin, Teacher, Student), and permissions.
  + Integrate with a secure password management system.
* **Frontend:**
  + Design and develop user-friendly registration and login interfaces.
  + Implement role-based access control to restrict access to specific features and functionalities.
  + Create user profiles for managing personal information and preferences.

**2. Organization and Class Management**

* **Backend:**
  + Develop the core organizational structure, including the creation and management of organizations, classes, and sections.
  + Design a flexible database schema to accommodate various organizational structures.
* **Frontend:**
  + Create intuitive interfaces for creating and managing organizations, classes, and sections.
  + Implement a clear visual representation of the organizational hierarchy (e.g., tree view).

**3. Subject Mapping and Teacher Assignment**

* **Backend:**
  + Develop functionalities for creating subjects and mapping them to specific classes and sections.
  + Implement mechanisms for assigning teachers to subjects and classes.
  + Create a database schema to store subject-teacher associations.
* **Frontend:**
  + Develop user-friendly interfaces for mapping subjects to classes and sections.
  + Implement clear visual representations of subject-teacher assignments.
  + Allow for flexible assignment of multiple teachers to a single subject or class.

**4. Evaluation Workflow and Basic Tools**

* **Backend:**
  + Develop the core evaluation workflow, including the ability to select classes, subjects, and access assignments.
  + Implement secure storage mechanisms for submitted assignments (e.g., file uploads, integration with learning management systems).
* **Frontend:**
  + Design and develop a user-friendly interface for teachers to access and evaluate assignments.
  + Implement basic annotation tools (e.g., text highlighting, comments, basic grading).
  + Provide a mechanism for teachers to provide feedback to students.

**5. Reporting and Basic Analytics**

* **Backend:**
  + Develop basic data processing and export functionalities (e.g., Excel, CSV).
  + Implement basic calculations for analytics (e.g., average scores, highest/lowest marks).
* **Frontend:**
  + Provide basic data export options.
  + Display basic analytics in a user-friendly format (e.g., tables, simple charts).

**6. User Interface and User Experience (UI/UX)**

* **Design:**
  + Conduct thorough user research and needs analysis.
  + Design a visually appealing and intuitive user interface.
  + Create wireframes and prototypes for all key features.
* **Development:**
  + Implement the designed UI/UX elements.
  + Ensure the system is user-friendly and accessible to all users.

**7. Testing and Quality Assurance**

* **Unit Testing:** Conduct thorough unit testing of individual components and modules.
* **Integration Testing:** Test the integration of different components and modules.
* **User Acceptance Testing (UAT):** Involve end-users in testing the system and gathering feedback.
* **Bug Fixing and Refinement:** Continuously address bugs and refine the system based on testing results and user feedback.

**8. Deployment and Maintenance**

* **Deployment:** Plan and execute the deployment of the system to the production environment.
* **Maintenance:** Establish ongoing maintenance procedures, including bug fixes, security updates, and performance optimization.
* **Documentation:** Create comprehensive documentation for developers and users.

**Development Approach**

* **Agile Methodology:** Adopt an agile development approach, such as Scrum, to facilitate iterative development, flexibility, and rapid response to changes.
* **Prioritization:** Prioritize features based on user needs, business value, and development effort.
* **Continuous Integration and Continuous Delivery (CI/CD):** Implement CI/CD practices to automate the build, test, and deployment processes.