**Overall Plan**

Phase 1: Initial Setup and Core Features

1. **Environment Setup**: Create a virtual environment and initialize a Django project.
   * **Technologies**: Python, Django
2. **Database Design**: Define the database schema to store user information, course modules, progress, etc.
   * **Technologies**: SQLite, Django models
3. **Authentication and Authorization**: Implement the login and sign-up features.
   * **Technologies**: Django authentication, django-allauth for social login
4. **User Types**: Create different user types (Student, Teacher, Parent).
   * **Technologies**: Django user model inheritance
5. **Basic User Interface**: Create basic templates for each type of user.
   * **Technologies**: HTML, CSS, Bootstrap
6. **Welcome Page**: Implement the welcome page features (Login, Sign Up, Exit).
   * **Technologies**: Django views, templates
7. **Menu Page**: Implement the menu page with the relevant options for each type of user.
   * **Technologies**: Django views, templates

Phase 2: Educational Content and Progress Tracking

1. **Learning Modules**: Create the modules page and allow users to select different learning modules.
   * **Technologies**: Django views, templates, possibly a third-party service for video tutorials
2. **Challenges and Quizzes**: Implement the quiz and challenge functionality.
   * **Technologies**: Django forms, JavaScript for interactivity
3. **Progress Tracker**: Implement the progress tracker and badges feature.
   * **Technologies**: Django models to store progress, JavaScript for progress bars

Phase 3: Teacher and Parent Features

1. **Teacher's Dashboard**: Create a dashboard for teachers to view progress reports.
   * **Technologies**: Django admin customization or separate views
2. **Parent's Dashboard**: Similar to the teacher's dashboard but with limited features.
   * **Technologies**: Django views, templates
3. **Feedback System**: Implement a feedback system for all types of users.
   * **Technologies**: Django forms, models to store feedback

Phase 4: Final Touches and Deployment

1. **Settings Page**: Implement the settings page where users can change various preferences.
   * **Technologies**: Django forms, views
2. **Testing**: Conduct thorough testing of all features.
   * **Technologies**: Django's test framework, Selenium for end-to-end tests
3. **Deployment**: Deploy the web application.
   * **Technologies**: Web server (e.g., Gunicorn, uWSGI), reverse proxy (e.g., Nginx), cloud provider (e.g., GCP)
4. **Monitoring and Maintenance**: Monitor the application and make updates as necessary.