

Learnsync Project Work Plan

1. Project Setup and Configuration

- ☐ Set up a new Django project.
- ☐ Create and configure the virtual environment.
- ☐ Install and configure necessary dependencies (Django, Stripe library, M-Pesa library, etc.).
- ☐ Create a new Django app for the project.

2. User Authentication and Registration

- ☐ Implement user authentication and registration functionality using Django's built-in authentication system.
- ☐ Add registration form and views for user sign-up.
- ☐ Set up login and logout functionality.
- ☐ Create user profile pages.

3. Course Management

- ☐ Create models for courses, lessons, and materials.
- ☐ Implement CRUD (Create, Read, Update, Delete) functionality for courses, lessons, and materials.
- ☐ Build views, templates, and forms for course management.
- ☐ Implement search functionality to allow users to find courses.

4. Payment Gateway Integrations

- ☐ Research and integrate the M-Pesa API or SDK into the project.
- ☐ Implement M-Pesa payment processing functionality for course purchases.
- ☐ Create views and templates for M-Pesa transactions.
- ☐ Handle transaction status callbacks and update user accounts accordingly.
- ☐ Research and integrate the Stripe API or SDK into the project.
- ☐ Implement Stripe payment processing functionality for course purchases.
- ☐ Create views and templates for Stripe transactions.
- ☐ Handle transaction status callbacks and update user accounts accordingly.

5. User Interaction and Collaboration

- ☐ Implement user-to-user messaging functionality.
- ☐ Build views, templates, and forms for messaging.
- ☐ Allow users to collaborate on course materials through comments or discussions.
- ☐ Implement a rating/review system for courses.

6. User Dashboard and Progress Tracking

- ☐ Create a user dashboard page to display enrolled courses, progress, and completed courses.
- ☐ Implement functionality to track user progress within a course.
- ☐ Add features for bookmarking and saving favorite courses.

7. Tweaks and Enhancements

- ☐ Implement additional features or tweaks as required, such as notifications, recommendations, etc.
- ☐ Improve the user interface and user experience.
- ☐ Conduct thorough testing and bug fixing.
- ☐ Optimize the performance of the application.
- ☐ Ensure code quality and adhere to best practices.
- ☐ Deploy the application to a suitable hosting platform.

8. Documentation and Deployment

- ☐ Document the project's setup instructions, usage guidelines, and API documentation.
- ☐ Write tests for critical functionalities and document the testing process.
- ☐ Prepare the project for deployment to a production environment.
- ☐ Deploy the application to a production server or cloud platform.
- ☐ Set up monitoring and error tracking tools for the deployed application.