# Project Roadmap & Role Assignment for miniMoodle Development

The demand for online educational platforms has surged in recent years, driven by the need for flexible, scalable, and cost-effective learning solutions. Many existing platforms, such as Moodle and Canvas, are powerful but often too complex or resource-intensive for small-scale educational institutions and individual educators.

This creates a gap in the market for a lightweight and efficient alternative.

## 1. Overview of Roles

For efficient development, the project is divided into frontend and backend roles.

• Frontend Developer – Касым Тимур: Responsible for UI/UX, dynamic interactions, and API integration.

• Backend Developer – Акылбек Мендибаев: Responsible for server logic, database management, and API implementation.

## 2. Project Plan with Role-Based Task Assignment

### Phase 1: Project Setup & Environment Configuration

**Backend (Акылбек Мендибаев)**

* Set up Go project structure.
* Configure PostgreSQL database.
* Adding roles to users(admin, teacher, student roles)
* Set up RESTful API using Gorilla Mux.

**Frontend (Касымбеков Тимур)**

* Initialize frontend project (HTML/CSS/JavaScript).
* Create project structure and install dependencies.
* Set up basic routing for pages (Home, Course List, Lesson Details).

### Phase 2: User Management & Authentication

**Backend (Акылбек Мендибаев)**

* Implement API endpoints for user registration & authentication (/api/register,

/api/login).

* Implement user authentication system.
* Secure authentication using JWT.
* Create a role-based authorization system (admin, teacher, student).

**Frontend (Касымбеков Тимур)**

* Design login & registration pages.
* Implement authentication flow using JWT.
* Set up protected routes based on user roles.

### Phase 3: Course & Lesson Management

**Backend (Акылбек Мендибаев)**

* Implement APIs for course and lesson management (/api/courses, /api/lessons).
* Set up relationships between courses, lessons, and users.
* Develop CRUD operations for lessons (create, read, update, delete).

**Frontend (Касымбеков Тимур)**

* Build UI for course listing and lesson details.
* Implement dynamic rendering of lessons.
* Set up interaction with backend for lesson creation and management.

### Phase 4: Assigning Teachers & Students

**Backend (Акылбек Мендибаев)**

* Implement API for assigning teachers to lessons (/api/lesson/assign-teacher).
* Implement API for assigning students to lessons (/api/lesson/assign-students).
* Create a database table for managing lesson-student relationships.

**Frontend (Касымбеков Тимур)**

* Build modal UI for assigning teachers and students.
* Fetch teachers and students dynamically from backend.
* Implement assignment logic with API integration.

### Phase 5: Grading System

**Backend (Акылбек Мендибаев)**

* Implement API for assignment creation (/api/quiz).
* Develop automated grading system.

**Frontend (Касымбеков Тимур)**

* Create UI for assignments.
* Implement automated grading display.

### Phase 6: Testing, Deployment & Final Touches

**Backend (Акылбек Мендибаев)**

* Perform backend testing (Postman, unit tests).
* Optimize database queries and API responses.
* Deploy backend on AWS or Google Cloud.

**Frontend (Касымбеков Тимур)**

* Perform frontend testing (browser compatibility, responsiveness).
* Optimize UI performance.
* Deploy frontend on Vercel or Netlify.

## 3. Tools & Technologies

**Backend**: Go, PostgreSQL, Gorilla Mux, JWT, AWS/GCP

**Frontend**: React, HTML, CSS, JavaScript, Axios

**Version Control:** GitHub/GitLab

**Project Management**: Trello

## 4. Expected Deliverables

• Fully functional miniMoodle system with user authentication.

• Role-based access for Admins, Teachers, and Students.

• Course & lesson management system.

• Student & teacher assignment system.

• Grading, communication, and feedback features.

• Cloud deployment with documentation.

## 5. Summary of Responsibilities

|  |  |  |
| --- | --- | --- |
| Task | Касым Тимур (Frontend) | Акылбек Мендибаев (Backend) |
| UI/UX Development | Yes | No |
| API Integration | Yes | Yes |
| Database Management | No | Yes |
| User Authentication | Yes | Yes |
| Course & Lesson Management | Yes | Yes |
| Assigning Teachers & Students | Yes | Yes |
| Grading System | Yes | Yes |
| Deployment | No | Yes |

## Conclusion

With this roadmap, miniMoodle will be a lightweight, user-friendly, and scalable learning platform.

The successful implementation of miniMoodle will:

* Provide educators with a lightweight and cost-effective LMS tailored to their needs.
* Enhance the learning experience for students through an intuitive and accessible platform.
* Reduce administrative overhead for institutions by automating routine tasks.
* Establish miniMoodle as a scalable foundation for future enhancements and integrations.