# Project 2: Course Platform (Proy2_Cursos)

Relevant source files

## Purpose and Scope

This document covers Project 2 within the IronHack Course 2 monorepo - a course management platform built with Express.js and MySQL. The system implements user authentication, session management, and course administration features. It represents an intermediate-level web application that builds upon basic CRUD concepts while introducing ORM usage and advanced authentication patterns.

For information about the overall repository structure and development environment, see Overview. For details about the database models and ORM configuration, see Data Models & Database. For authentication implementation details, see Authentication & Middleware.

## System Architecture Overview

The course platform follows a traditional MVC architecture pattern with Express.js as the web framework, EJS for templating, and session-based authentication with JWT tokens.

### Application Structure


Sources: Proy2_Cursos/index.js 1-21 Proy2_Cursos/config/main_config.js 1-26

## Core Configuration

The application configuration is centralized in `main_config.js` which sets up the Express application with essential middleware and templating engine.

### Express Application Setup

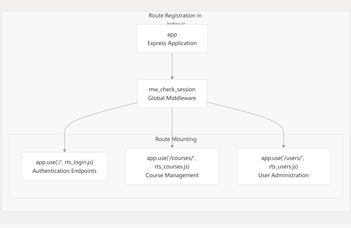

The session configuration uses a hardcoded secret and 60-second timeout as defined in Proy2_Cursos/config/main_config.js 22

Sources: Proy2_Cursos/config/main_config.js 6-24

## Routing Architecture

The application implements a modular routing system where each functional area has its own router module.

### Route Registration Flow



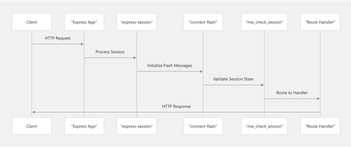

The `mw_check_session` middleware is applied globally before any route handling, ensuring session validation occurs for all requests as shown in Proy2_Cursos/index.js 12

Sources: Proy2_Cursos/index.js 12-18

## Session and Middleware Integration

The application uses a layered approach to session management and request processing.

### Middleware Processing Chain




The session middleware is configured with a simple secret `'keyboard cat'` and a 60-second cookie timeout, indicating this is a development configuration as seen in Proy2_Cursos/config/main_config.js 22

Sources: Proy2_Cursos/config/main_config.js 14 Proy2_Cursos/config/main_config.js 22 Proy2_Cursos/index.js 6 Proy2_Cursos/index.js 12

## Module Dependencies and Exports

The application follows a modular export pattern that enables clean separation of concerns and dependency injection.

### Configuration Module Exports

| Export | Type | Purpose |
|---|---|---|
| `app` | Express Application | Main Express instance with configured middleware |
| `router` | Express Router | Router instance with merged parameters |
| `session` | Session Module | Express-session module reference |

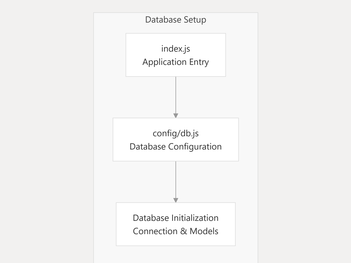
The main application entry point imports the configured Express app and router from the config module, then applies global middleware and mounts route handlers as shown in Proy2_Cursos/index.js 5

Sources: Proy2_Cursos/config/main_config.js 25 Proy2_Cursos/index.js 5

## Database Integration

The application initializes database connectivity through a dedicated database configuration module that is required during application startup.

### Database Initialization Flow




The database module is required early in the application lifecycle as shown in Proy2_Cursos/index.js 9 ensuring database connectivity is established before route handling begins.

Sources: Proy2_Cursos/index.js 9