# Intro

This is documentation for the Al-Safa LMS project, provided by FSH (Fratello Software House) to describe a technical overview for the project.

# Project Overview

Learning Management System (LMS) that manages all operations, processes, contents …etc. for Al-Safa Academy.

Based on our meeting, the following were the existing feature the system has:

* Users
* Content Management
* E-Learning
* Exams and Assignments
* Advertisements
* Certifications
* Academic year, Semesters, Subjects …etc.
* Integration with payment getaway
* Notifications
* Reports and Statistics
* Blogs

**New features that we can add:**

* New UI/UX
* New Control Panel: this will be available for administrator to manage data.
* Purchase Courses
* Content Management system and Landing page.
* Certificates Management  
  All LMSs should allow for users to receive a certificate after completing a course.
* Course Management
  + Easily create courses.
  + Use existing templates to make creating new courses simpler.
  + Enroll users.
  + Categorize courses.
  + Monitor and track users’ progress.
* Multi Languages (Arabic, English …etc.): Localization is the process of translating course material into a different language.
* User Management (Enhance users, roles, permissions …etc.)

## Used Technologies

This project was built using .NET Core (MVC) and these technologies:

* DB: MS SQL Server.
* Third party integration with Microsoft Teams
* Online payment integration with Senang Pay
* Tools: Visual Studio, MS SQL Server, Microsoft SQL Server management studio.

## Installation

System Requirements

Before installing the LMS, ensure your system meets the following minimum requirements:

* Operating System: Windows Server
* Web Server: Internet Information Services (IIS)
* Database: MSSQL

Section 2: Configuration

Basic Settings

* Site Name: This will be the name of your LMS site.
* Site Language: Multi-language.
* Time zone: Select your appropriate time zone.

## Latest Code and Data

Latest changes will be sent as Zip file.

Latest database copy is already in the production server.

## Usage:

For Administrators

* User Management: create, edit, and delete user accounts. Manage user roles and permissions.
* Course Management: Create and categorize courses, manage semester, course content, course prerequisite, manage enrollments, etc.
* Exam Management: Create Bank of questions, create template of exam, shuffle exam,
* CMS: to create, edit, and delete (Blogs, New), manage sliders, manage static page like about us, Teams, What people saying,
* Calendar: create, edit, and delete calendar like public holidays
* Reports: Student, Courses, student with courses, payments, logs, etc.

For Instructors

* Student Management: Monitor student, Attendances
* Course Management: Course content.
* Exam Management: Create exam, assignment, view answers of exam, exam correction.

For Students

* Course Enrollment: Learn how to search for and enroll in courses.
* Course: View units, lessons and resources.
* Assignments and Exams: Learn how to submit assignments and take Exams.
* Certificates: View certificates
* Calendar: View public holiday, courses hours and exams, assignments

For Guest

* CMS: View blogs, news, teams, about us and contact us.
* Register: allow to create account as student.
* Newsletters: subscribe newsletters.

# Tables and DB Structure

In this section we will show you an overall description of the database and tables structure for the core flow of the project.

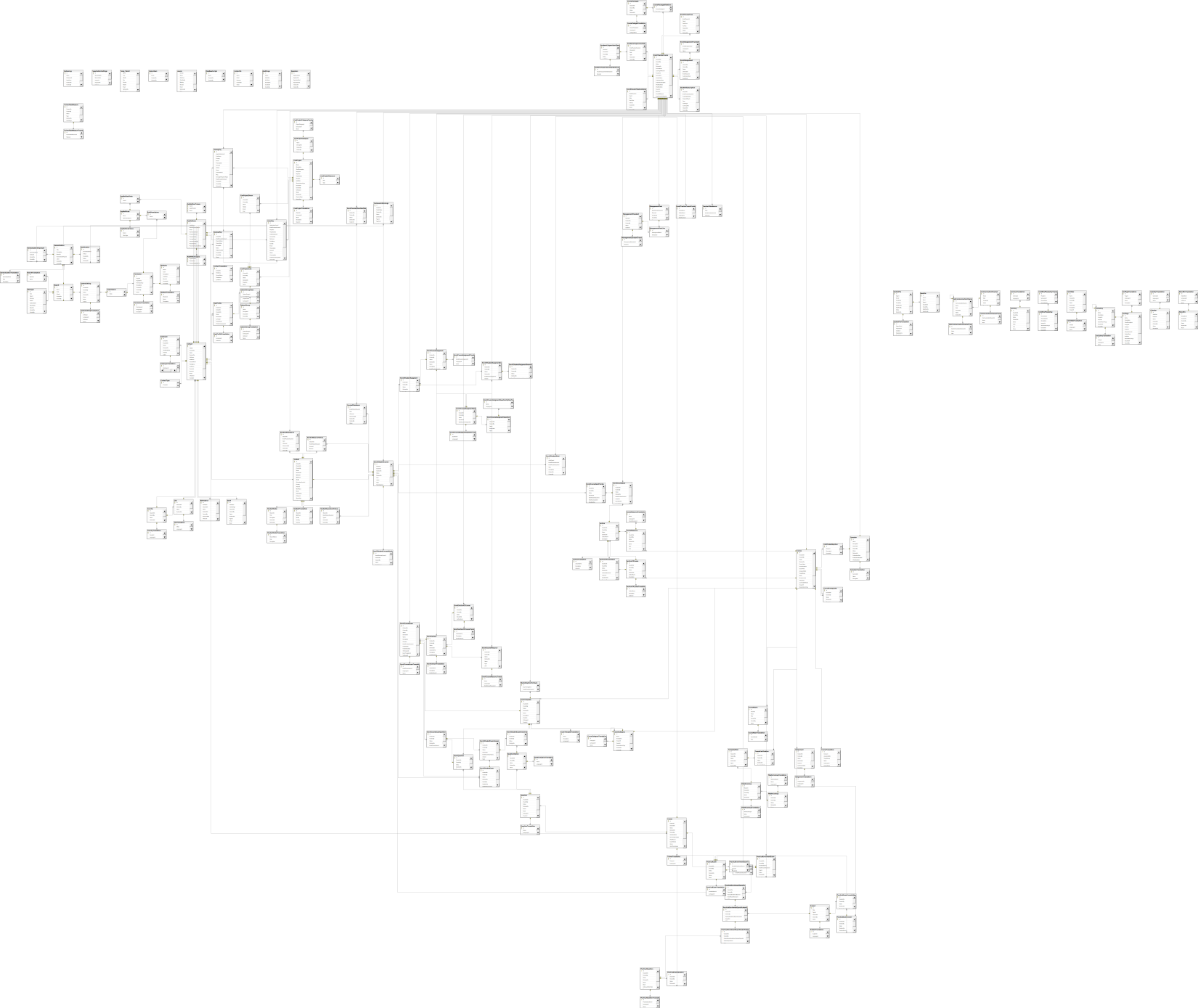
We build the database tables in clear way with clear tables name and columns name so anyone can understand what each table contains. For example, Lecture table named with Lecture and as most of the tables it contains shared columns as follows:

* Id (Auto increment integer, the primary key of the table)
* CreatedOn (date and time of creation)
* CreatedBy (User who created the record)
* Status (Active, Deactivate , Deleted, … etc.)
* DeletedOn (Date and time of deletion the record)

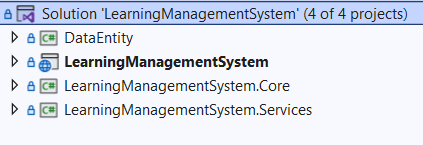
# 

# Database Diagram:

Diagram will be sent separately in high resolution.



# Project Structure:



The solution contains the following projects:

* Web Project (Learning Management System)

This built on MVC design pattern (Standard from Microsoft)

* Core Project

Contains Enums, Extension methods and constants

* Services

This contains all the real logic and connection with database tables.

* Data Entity

This contains the database models and the view models.

# App & Permissions

As an admin, you can define multiple roles and permissions for other admins to use.

These roles can be found in the URL "/ControlPanel/AspNetRoles". and permissions can be associated with these roles in "/ControlPanel/Permissions".

To define a new role for user type <new-type>: we do the followings:

* Add a new role from the URL "/ControlPanel/AspNetRoles".
* Go to the Role Details that will view all permissions in the system and the admin can select permission for this role.
* Go to the Contact page in the URL "/ControlPanel/UserProfiles" and add a new employee and select the role for this employee or you can edit the role for any employee in the same page.
* Note: The user can take many roles.
* Note: don’t change the permission key or page name without changing it from the code. Because we write the permission key and page name as a hardcode in all controllers. I prefer to hide this page from the client to avoid deleting permissions and cause problems.

# System Admin Credentials

* Administrator.
  + Admin can add, delete and view everything in the system.
  + He can Also reset passwords for each user from the URL "/ControlPanel/UserProfiles" for each user having an operation to reset his password.

# Deployment of Project

To deploy the project with your latest changes, we recommend using Visual Studio. Use the following steps to deploy:

1. At first you need to be sure that all functions work fine (No build errors)
2. Go to the web project and right click of the mouse then publish.
3. After publishing you will download the publish files on your device.
4. Be sure to delete the appsetting file and document folder before moving them to the server.
5. Go to the client server then stop the site then copy the files to the previous files then they will be replaced.
6. Note: if there's no new packages you can copy only the changed files like (DLL, CSS, JS …etc)