Project report on

**Mobile Application for IIT**

A project submitted  
By  
Shadman Sakib  
ID: 181909, Batch: MIT19

Under the Supervision of

**Dr. Kazi Muheymin-Us-Sakib**ProfessorInstitute of Information TechnologyUniversity of Dhaka



Institute of Information Technology  
University of Dhaka

4th January, 2020

January 2, 2020

To

Dr. Kazi Muheymin-Us-Sakib   
Professor  
Institute of Information Technology   
University of Dhaka.

**Subject: Letter of submission of project report.**

Sir,

I would like to submit my project report titled “Mobile Application for IIT”, as a part of the Master in Information and Technology curriculum. I would also take this opportunity to thank you for helping me throughout the project.

I have devoted my utmost dedication to make this report an informative one. I hope you will find it to be objective, systematic and reliable. Please feel free to contact me if you have any inquiries, I would be very happy to provide you with any clarification regarding the report.

Thank you again for the wonderful opportunity.

Regards,

Shadman Sakib  
Roll: 181909  
Mobile: 01770247346  
Email: sakib.buet51@gmail.com

Declaration of Original Work

I declare that the Project entitled **Mobile Application for IIT**, which I submitted to the faculty of the **Institution of Information Technology, University of Dhaka** is my own work. To the best of my knowledge, it does not contain materials published or written by another person, except where due citation and acknowledgement is made in my project documentation. The contributions of other people whom I worked with to complete this project are explicitly cited and acknowledged in my project documentation.

I also declare that the intellectual content of this project is the product of my own work. I conceptualized, designed, encoded, and debugged the source code of the core programs in my project. The source code of third party APIs and library functions used in my program are explicitly cited and acknowledged in my project documentation. Also duly acknowledged are the assistance of others in minor details of editing and reproduction of documentation.

In my honor, I declare that I did not pass of as my own the work done by another person. The only person encoded the source code of my programs is me. I understand that I may get a failing mark if the source code of my program is in fact the work of another person.

Shadman Sakib  
Roll-MIT181909

This declaration is witnessed by:

Dr. Kazi Muheymin-Us-Sakib  
Professor, IIT  
University of Dhaka

**ACKNOWLEDGEMENTS**

A lot of effort and study have been put to make this report into a reality. This would not have been possible without the genuine support and assistance provided by the people whom I approached during the various stages of writing this report. Firstly I would like to express my gratitude to my academic supervisor Dr. Kazi Muheymin-Us-Sakib, Professor, Institute of Information Technology, University of Dhaka; for his advice, counseling, direction and help. Without his guidance this report would not have been possible.

I am honored to convey my cordial thanks and gratefulness to Dr. Md. Shariful Islam, honorable Director, Institute of Information Technology, for his precious encouragement.

I am very much grateful to Dr Zerina Begum, Professor, Coordinator – MIT-19, Institute of Information Technology, University of Dhaka for her consistent encouragement and inspiration.

I also must thank to all the faculty members of Institute of Information Technology for all their direct and indirect encouragement and assistance in this work. Their cooperation, suggestion, guidance and sincere encouragement played significant role throughout my working period.

**Abstract**

The availability of low-cost mobile phone and always connected to internet through GSM or WIFI has made a huge opportunity to provide Information and communication Technologies to the User. The project has been selected to develop a mobile application which would solely be used for IIT students for the purpose of managing their class-routines, scheduling class test and examination and other academic events. The cross-platform mobile application resulting would solely be manageable by the students without regular intervention from the admin panel. However its also possible to manage and complete the task from admin panel that can be done by students from the mobile application.

**SYNOPSIS:**

Title: Mobile Application for IIT

Student’s Name: Shadman Sakib

Venue: IIT, DU

Duration: 2 months 28 days

Platform: Android, IOS

Application Software

Table of Contents

[1. Introduction 1](#_Toc28794843)

[1.1. Objective 1](#_Toc28794844)

[1.2. Scope 1](#_Toc28794845)

[2. Methodology 2](#_Toc28794846)

[2.1. System Analysis 2](#_Toc28794847)

[2.2. Functional Requirement 2](#_Toc28794848)

[2.3. Non Functional Requirements: 2](#_Toc28794849)

[2.4. Software Requirement 2](#_Toc28794850)

[2.5. Development Algorithm 3](#_Toc28794851)

[2.5.1. Shared Component 3](#_Toc28794852)

[2.5.2. Server 3](#_Toc28794853)

[2.5.3. Mobile Application 4](#_Toc28794854)

[2.6. Data Flow Diagram 5](#_Toc28794855)

[2.7. Entity Relation Diagram 6](#_Toc28794856)

[3. Implementation 7](#_Toc28794857)

[3.1. Programming Languages 7](#_Toc28794858)

[3.1.1. API Handlers 7](#_Toc28794859)

[3.1.2. Mobile 8](#_Toc28794860)

[3.1.3. Web Client 9](#_Toc28794861)

[3.1.4. Web Server 9](#_Toc28794862)

[3.2. Third Party Libraries. 9](#_Toc28794863)

[4. Result **Error! Bookmark not defined.**](#_Toc28794864)

[4.1. Server 10](#_Toc28794865)

[4.2. Mobile 13](#_Toc28794866)

[5. Conclusion 17](#_Toc28794867)

[6. References 18](#_Toc28794868)

**Table of Figures**

[Figure 01 Data Flow Diagram 5](#_Toc28795031)

[Figure 02 Entity Relation Diagram 6](file:///C:\Users\sakib\Desktop\MIT-181909%20(Recovered).docx#_Toc28795032)

[Figure 03 Admin Panel Dashboard 10](#_Toc28795033)

[Figure 04 Admin Panel Batches 10](#_Toc28795034)

[Figure 05 Admin Panel Batch Information 11](#_Toc28795035)

[Figure 06 Admin Panel Course Information 11](#_Toc28795036)

[Figure 07 Admin Panel Notices 12](#_Toc28795037)

[Figure 08 Admin Panel Students 12](#_Toc28795038)

[Figure 09 Mobile Student Login 13](#_Toc28795039)

[Figure 10 Mobile Dashboard 13](#_Toc28795040)

[Figure 11 Mobile All Semester And Courses 14](#_Toc28795041)

[Figure 12 Mobile Current Semester And Classes 14](#_Toc28795042)

[Figure 13 Mobile Course Information 15](#_Toc28795043)

[Figure 14 Mobile Course Create 15](#_Toc28795044)

[Figure 15 Mobile Notices 16](#_Toc28795045)

[Figure 16 Mobile GradeSheet 16](#_Toc28795046)

# Introduction

The student management system via Mobile application provides an interface with the student database via API interfacing. Hence registering, editing and maintain students can all be done from the single Application. Moreover, Course and Lesson can be created and assigned to Teachers and a batch of student, which gives the possibility to make farther interactions. Assignment, Quiz or any other events like extra class or even cancelling a pre routine class can be achieved easily. The application automatically generates the routine for the current user and shows the event highlighted.

The need for such application is such that one can easily get the reminder of today’s class and if any events in the coming days. As such, every student has its own username/login to interact with the database and get his/her own routine and events.

In this document, I try to describe the background, project scheduling, purpose, product scope and future work. Some main usage scenarios are described in more detail, and functional and non-functional requirements are provided.

## Objective

The prime objective is to develop a system where admin can create and manage students, courses, course materials. So of this features are extended to a client mobile application which would be used by students. The students can be benefitted by digital class routine, upcoming events and notices.

The application provides a huge opportunity to give instant notification about the events. Also the application generates the routine dynamically showing all the lessons and activities. The system also provides the ability to track course, lesson and grades.

## Scope

The software system will have two core components. One admin panel for user management and optionally other activities. And another is the cross platform mobile application which would be used by the students. Most the management can be offloaded to the student and hence maximizing the efficiency of the whole system.

More specifically the system has been developed to communicate between the students in a common platform sharing all their academic activities.

# Methodology

## System Analysis

This section analyze the requirements of the system

* To provide a system that makes the user registration and maintenance portable, user-friendly and vastly expandable.
* Admin can add new students via uploading files containing student information.
* The student will be notified via an email confirming the registration has been completed.
* The user would then install the application. Upon new registration, the student must complete registration confirming his/her mobile number and providing a new password.
* The student, if permitted can create new course, lesson and gradually fill-up his/her academic activities. For any new course enrollment, the student would be notified via push notification.
* The events can both be created from admin and student side. Admin can chose the scope of the event by selecting all batch or selective batch. The events created by a student would be exclusive to his/her batch mates.

## Functional Requirement

The system is expected to provide the following functionalities

* Keep track of the logged-in users via token system under the hood
* Provide routine for the logged in User
* Provide a mechanism to add/edit Information for the privileged user
* Broadcast a notification to the corresponding user as soon as its created
* Keep details of the Users, Course, and Lesson and present them to the appropriate user.

## Non Functional Requirements:

Nonfunctional requirements define system properties and constraints. It arise through user needs, because of budget constraints or organizational policies, or due to the external factors such as safety regulations, privacy registration and so on.

Following Non-functional requirements will be there to ensure comfortable user experience.

* Secure access of confidential data (User details).
* 24 X 7 availability.
* Flexible service based architecture will be highly desirable for future extension.

## Software Requirement

The server application is developed based on .net core 3.1 with MySQL database. The server application is able run from any .net core supported environment including Windows Azure Server, AWS hosting and Linux hosting. On a server enabled by proxy, the server has been tested to run behind nginx and apache. The software also implements a Docker file to integrate with Docker containers.

The cross platform mobile application is built on top of android and iOS. On android, minimum android version should be Android 5.0. And on iOS, the minimum supported iOS version is iOS 8.0. On both platform, a minimum of 4inch device screen with 480x600 is required to run the application.

## Development Algorithm

The core application is divided into two parts. The server side and the client side. The server contains the database and all the business logics. The client side is responsible for providing the user interface.

The client and server both depends on some common components which have been implemented in Shared Components.

### Shared Component

The core application is developed in C#. The shared project by the server and mobile would provide the common models and interfaces. This project is developed on C# .net standard language. This solution item includes all the Database models, API interfaces and other shared constants.

This part of the application does not contain any business logic or any data handlers. However the models define the relation between each other in this domain.

### Server

Server is responsible for creating Application Programming Interface(API) between the front side and the database. There are two solution item from server. The core responsibility of the IIT.Server are

* Bootstrap the server side application.
* Define all API endpoints.
* Consume the underlying database services to provide API.
* Manage Authentication and authorization through token. (JWT)
* Run a basic kestrel server on the predefined port from configuration.
* Contact with google server for Notification and Email.
* Handle any error and restart server side with previous state.

Though the Server.IIT hosts a client side web application, but it is not dependent on the client side any way. Currently the client side is a separate web application based on Blazor which provides an admin panel to the application based on the API provided by the server side. The core responsibility of the IIT.Client are

* Host the admin panel.
* Login and cache the current user and token.
* Authorize routes based on token.
* Provide a component based architecture which can be reused.
* The underlying engine works like a single page application. (SPA)

### Mobile Application

The mobile application is independent of any server side element. However it depends on the Shared project and inherits the model and API interfaces. The mobile application follows the MVVM (Model-View-ViewModel) design pattern and uses Microsoft Dependency Injection library to achieve a true Inversion of Control (IOC).

* The application is developed in Xamarin C#.
* The core application contains Views. A separate solution (Mobile.Core) implements all the viewmodel and required application services.
* The mobile project is shared by two separate solution, IIT.Android and IIT.IOS. The application provides an interface to provide cross platform dependency. The output would generate APK from Android side and IPA from IOS side.

## Data Flow Diagram

Data Flow Diagrams (DFD) is a way to represent system requirements in a graphical form which lead to modular design. A DFD describes what data flow (logical) rather than how they are processed, so it does not depend on hardware, software, data structure or file organization.

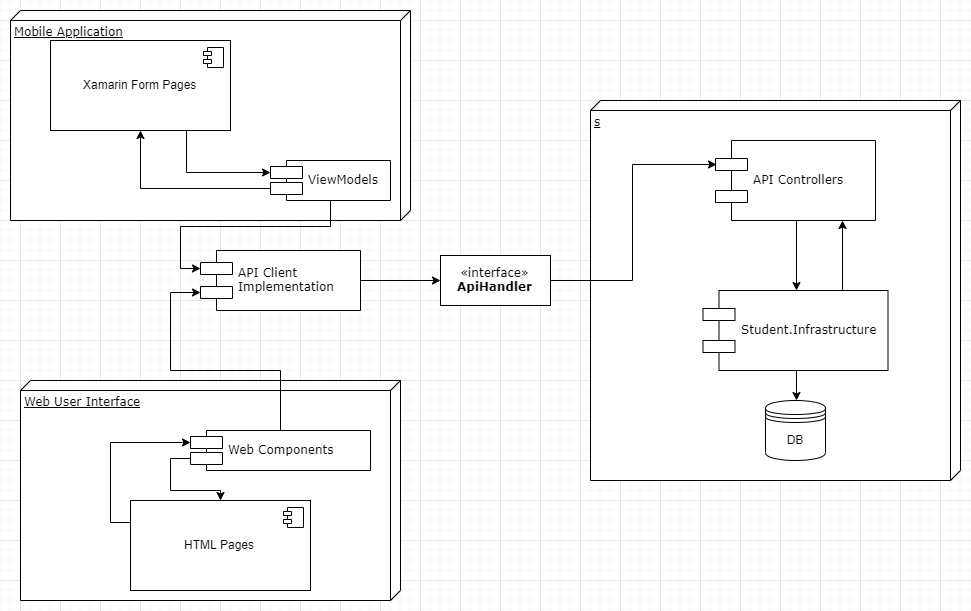


Figure 01 Data Flow Diagram

## Entity Relation Diagram

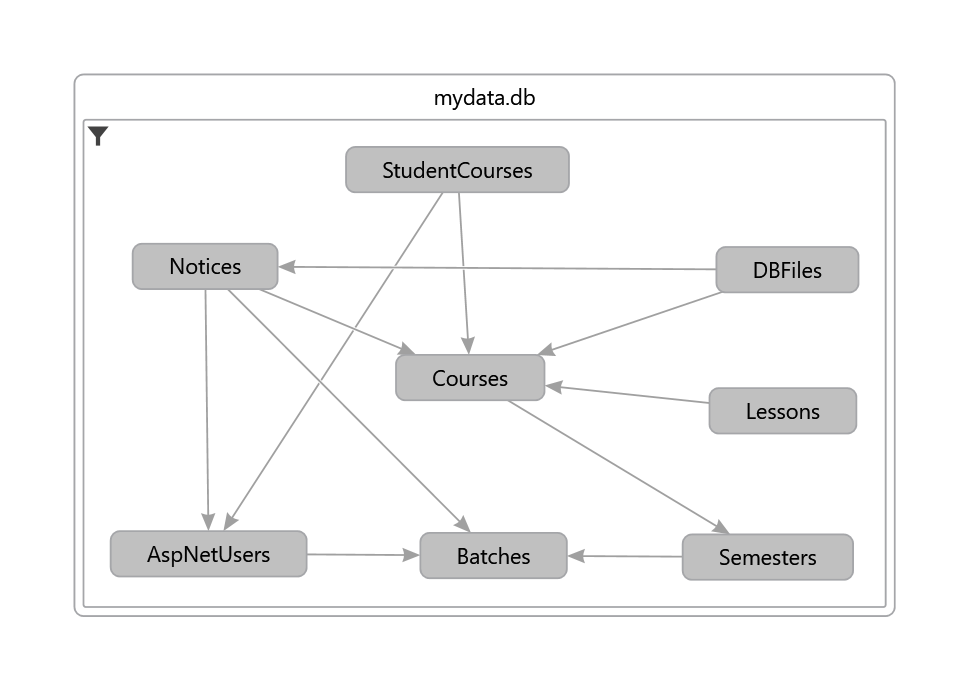


Figure 02 Entity Relation Diagram

# Implementation

## Programming Languages

The core system of the application is developed in C#. Other supporting languages used are XAML (Mobile View) and Html(Server). On the server side, Blazor client has been used which provides a good way to consume the C# language on the client side instead of JavaScript.

Visual Studio 2019 has been chosen for the core IDE of the system. The website has been debugged using the Chrome browser.

### API Handlers

The system provides some common API interface. On the client side, the api handlers are implemented to consume the underlying application. On the Server side, the API handlers implements the corresponding required services.

#### Client

Here, client is defined as both the mobile application and the web view that is used to render the Admin panel. This part is defined with the solution items “CoreEngine, Mobile.Core, MobileApp, IIT.Client”.

* The api handler implementation on the client side depends on a HttpWorker. HttpWorker is a wrapper around HttpClient. The core purpose of the HttpWorker is to send appropriate Get/Post request to the server. The base address of the Mobile application is defines in AppConstants.cs, whereas the base address of the IIT.Client is injected from server.
* The base api handler, an abstract engine contains several helper function. This class is dependent directly on the controller name of the server. The abstract engine enforces an easier way to send request to corresponding api endpoint. It takes the calling functions name, controller and constructs the api endpoint in the following way.
  + /api/[controllerName]/[methodName]
* The base api handler provides also a multipartUpload method which consumes a dbfile(File data) and convert into IFile for server side file consuming.
* Error handling is done In HTTPWorker.
* The corresponding api handlers are implemented
  + BatchEngine
  + CourseEngine
  + LessonEngine
  + NoticeEngine
  + MemberEngine

#### Server

The server defines the IIT.Server and Student.Infrastructure domain. The apiHandler interface are also implemented here to consume the service provided by Infrastructure layer. The interface management system enforces that the data type and parameters remain consistent throughout the system. So if the API Interface defines an api endpoint, the server must implement the endpoint.

The server is dependent on the Student.Infratructure to provide database access. All the services for accessing database are registered at startup and injected to the controller at runtime.

### Mobile

The mobile application is build using Xamarin.Forms. The core system uses a model-view-viewmodel structure. This application depends on the shared CoreEngine to consume the APIInterface and Models.

* In App.xaml the application has been bootstrapped with Microsoft dependency injection library to register the viewmodel, required service.
* MVVM navigation pattern has been used to navigate from one page to another. The Xamarin Pages are inherited from a custom page, which provides an opportunity to define the type of the viewmodel it will define on. The MVVM navigation is independent of the view.
* In testing, the service required by the viewmodels are required to be implemented in a test Service.
* The API interfaces are registered via a common Method. To reuse the caching technique of httpClient, the httpWorker is registered with singleton.
* The dependency injection takes care of the required service needed by the viewmodel. Each viewmodel has the lifetime of the view. However, each apiInterface has the lifetime of the Android Wake lifetime. That means, if the app is put to sleep, API handlers will be destroyed to release memory resources.
* The mobile application saves the token(JWT) in local storage.
* The token is used to login at startup if present. Otherwise, the login screen is showed.

### Web Client

This defines the IIT.Client which is responsible for rendering the admin panel. It is separated from the server application and can also be started separately. The client application is composes of C# and uses the Blazor Client technology.

* At startup, the application register the required services via Microsoft dependency injection.
* The web application is a component like structure. So each component works on its own. The required api services are injected at start.
* The component can request data from service via the api handlers on start or from user interaction.

### Web Server

This is the core server application. At startup it defines the api entry point. The web client application is also hosted within the server. (But can be started separately).

* The project depends on the student.infrastructure for database access. The base Services are registered using reflection at application start.
* The application start also initializes the databases implementing IhostService.
* The application creates a JWT token at successful login and manage authorization and authentication validating the token though out the API controllers.
* The current database used is MySQL. Depending on the deployment environment msSql, postgreSql, sqlite can also be used by defining them at startup.
* The student.infrastructure uses the object relation mapping defined in coreEngine to generate a database. The configuration is provided by the main startup file.

## Third Party Libraries.

The use of third party libraries have been kept as minimum as possible. However the following third party libraries have been used.

* NewtonSoftJson: Serialize object to string and vice versa.
* MatBlazor: Material theme provided for the WebClient application.
* MvvmLightStd: MVVM helpers for Mobile application.
* Jdenticon: Icon creator for mobile application.
* Blazored.Local: Provides local storage for C# on browser.
* Blazored.Inputfile: Provides Custom inputfile for C# on Browser.
* PancakeView: Provides gradient view on Xamarin Forms

# User Manual

The system output can be verified in two different module. One module running in server and can be visible via admin panel. And another is the mobile application running in a mobile environment.

## Server

The following result’s has been derived from Admin panel running on a windows machine.

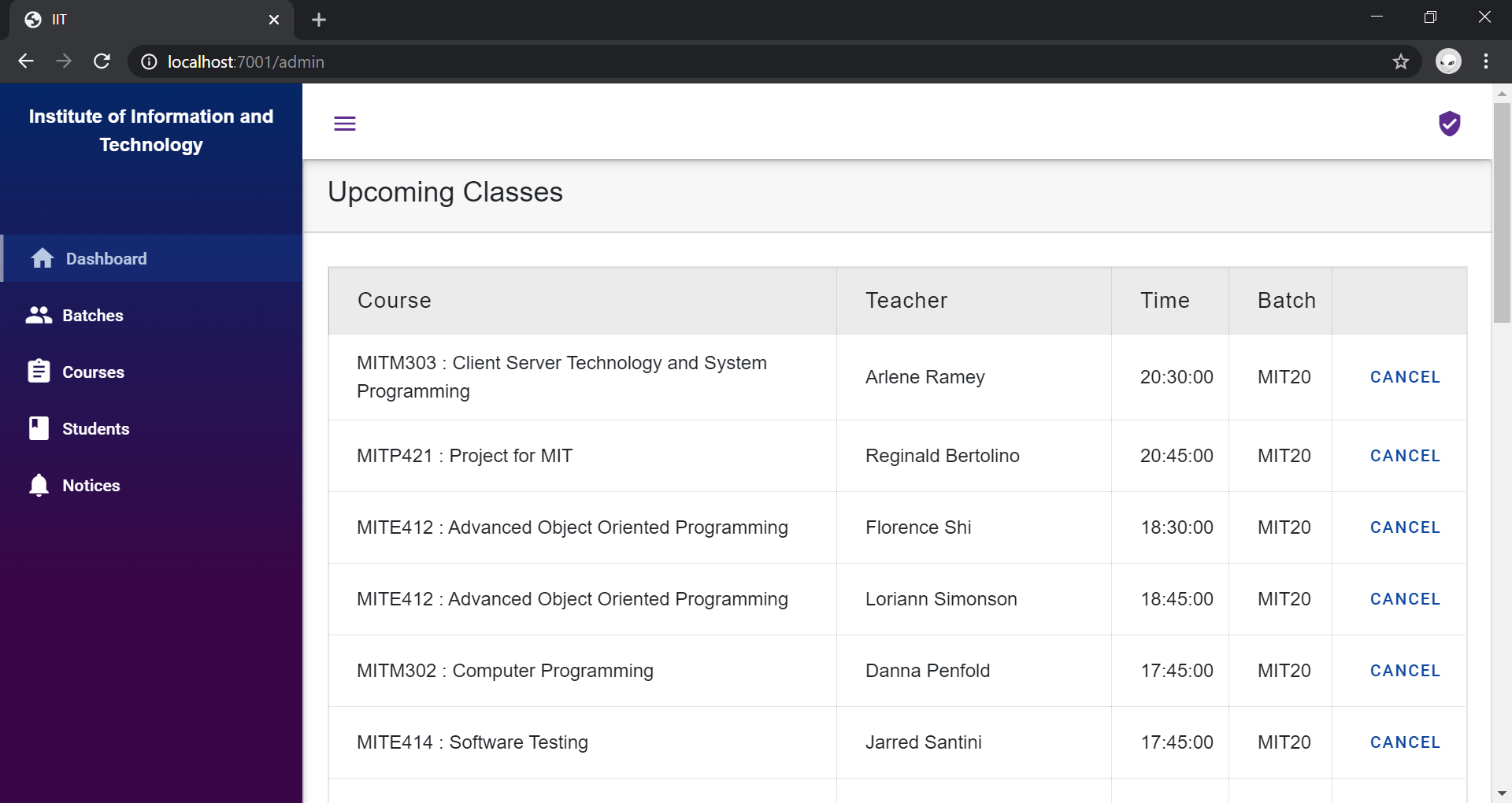


Figure 03 Admin Panel Dashboard

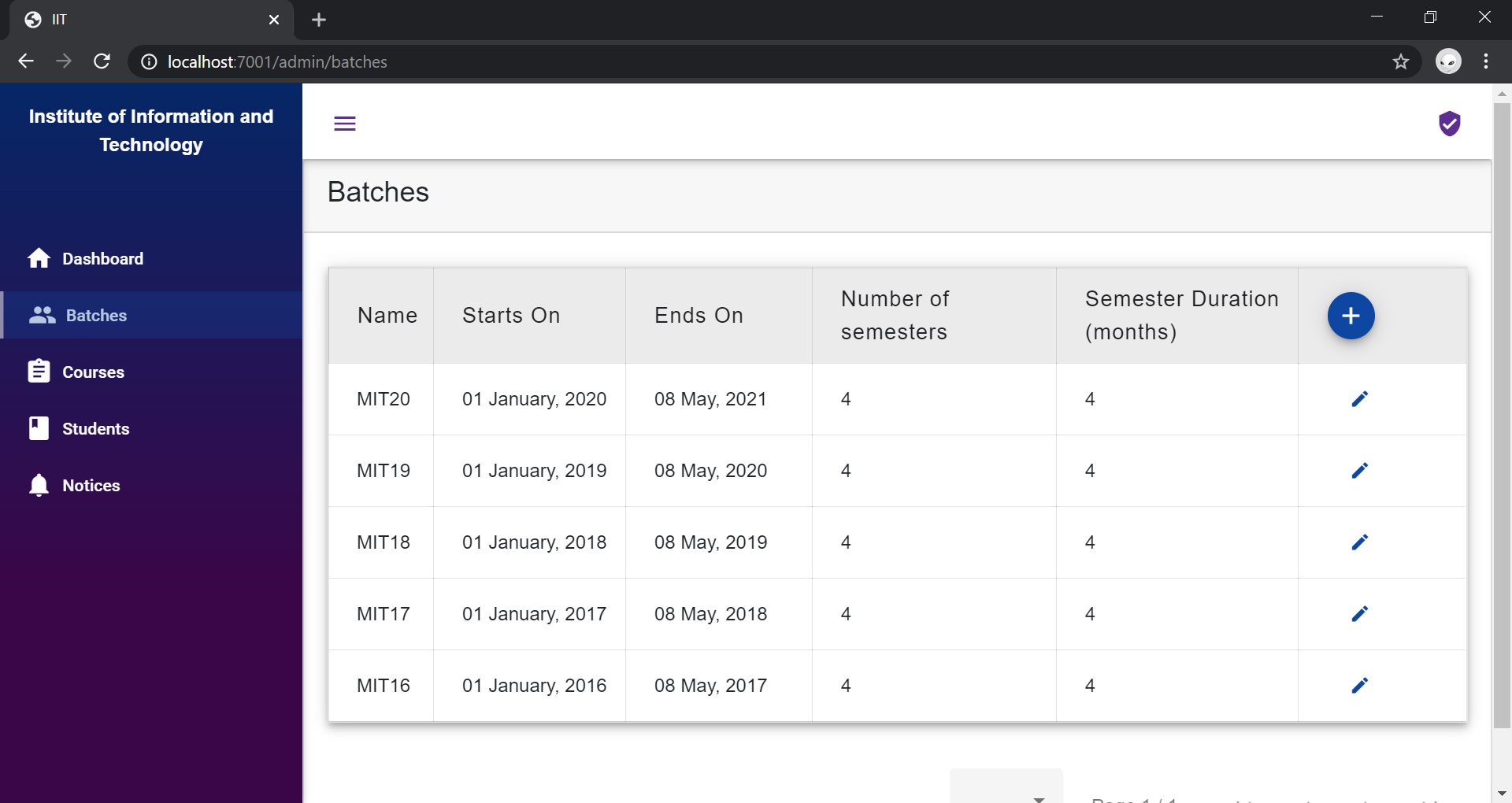


Figure 04 Admin Panel Batches

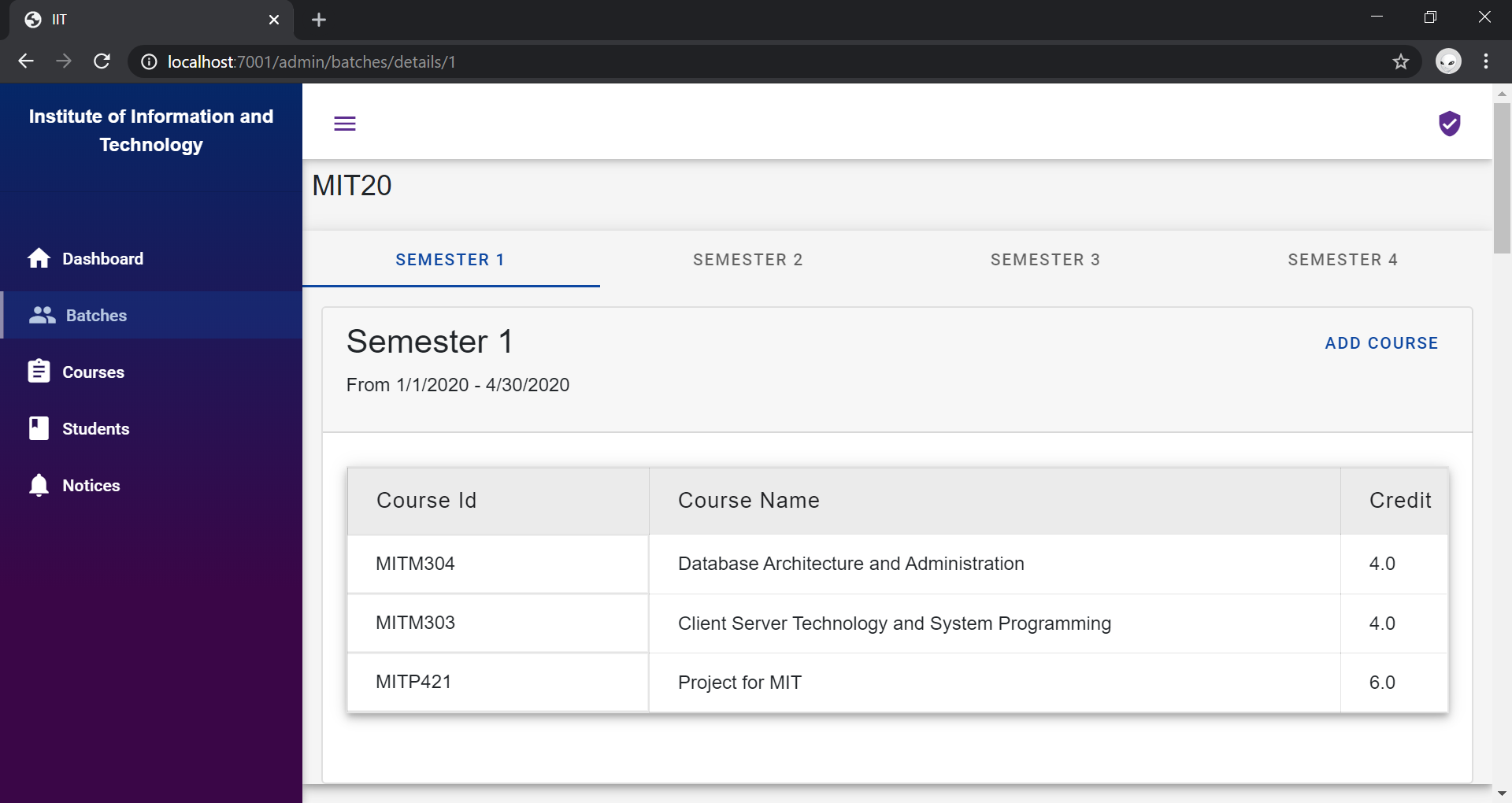


Figure 05 Admin Panel Batch Information

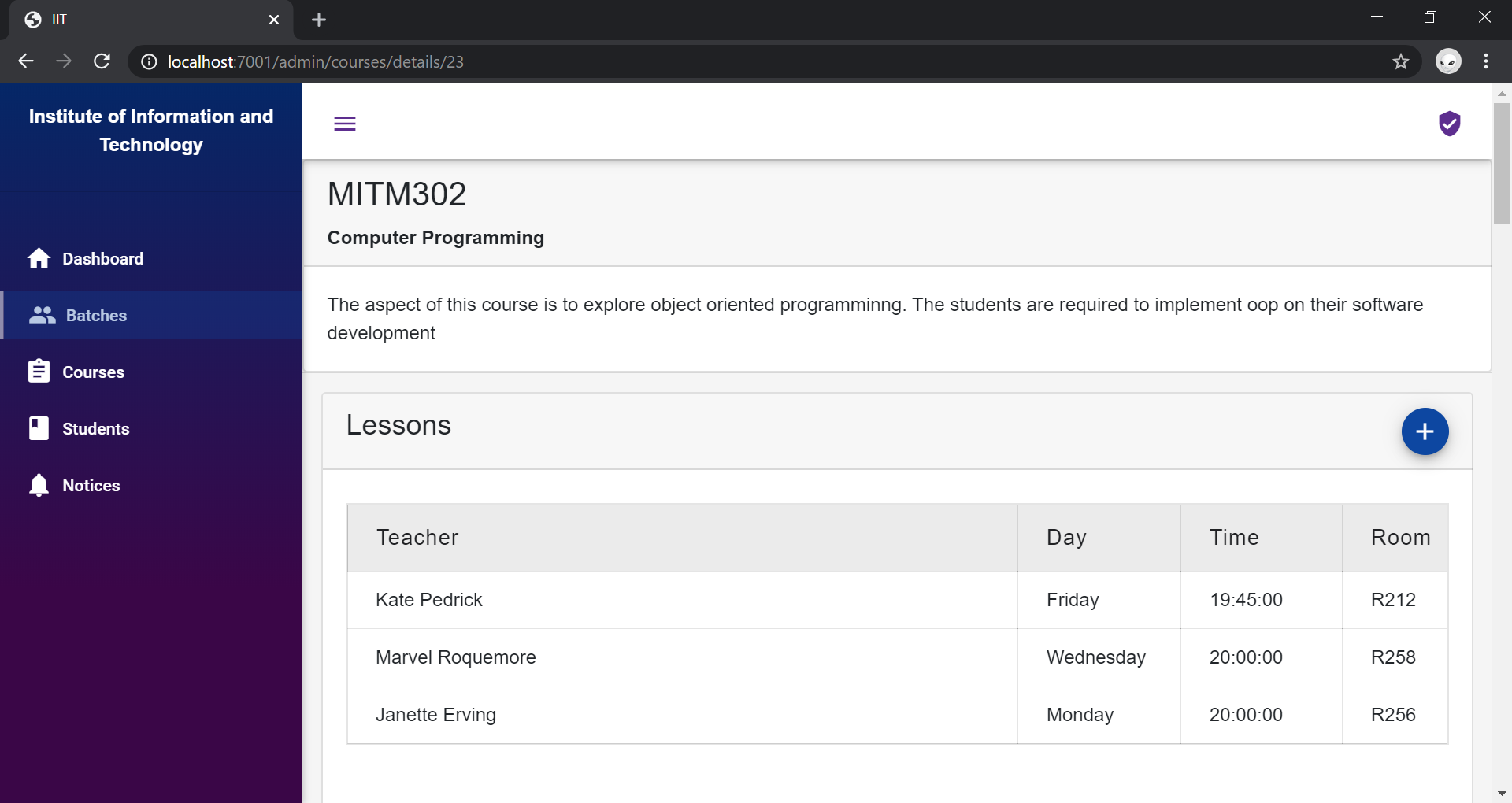


Figure 06 Admin Panel Course Information

Figure 07 Admin Panel Notices

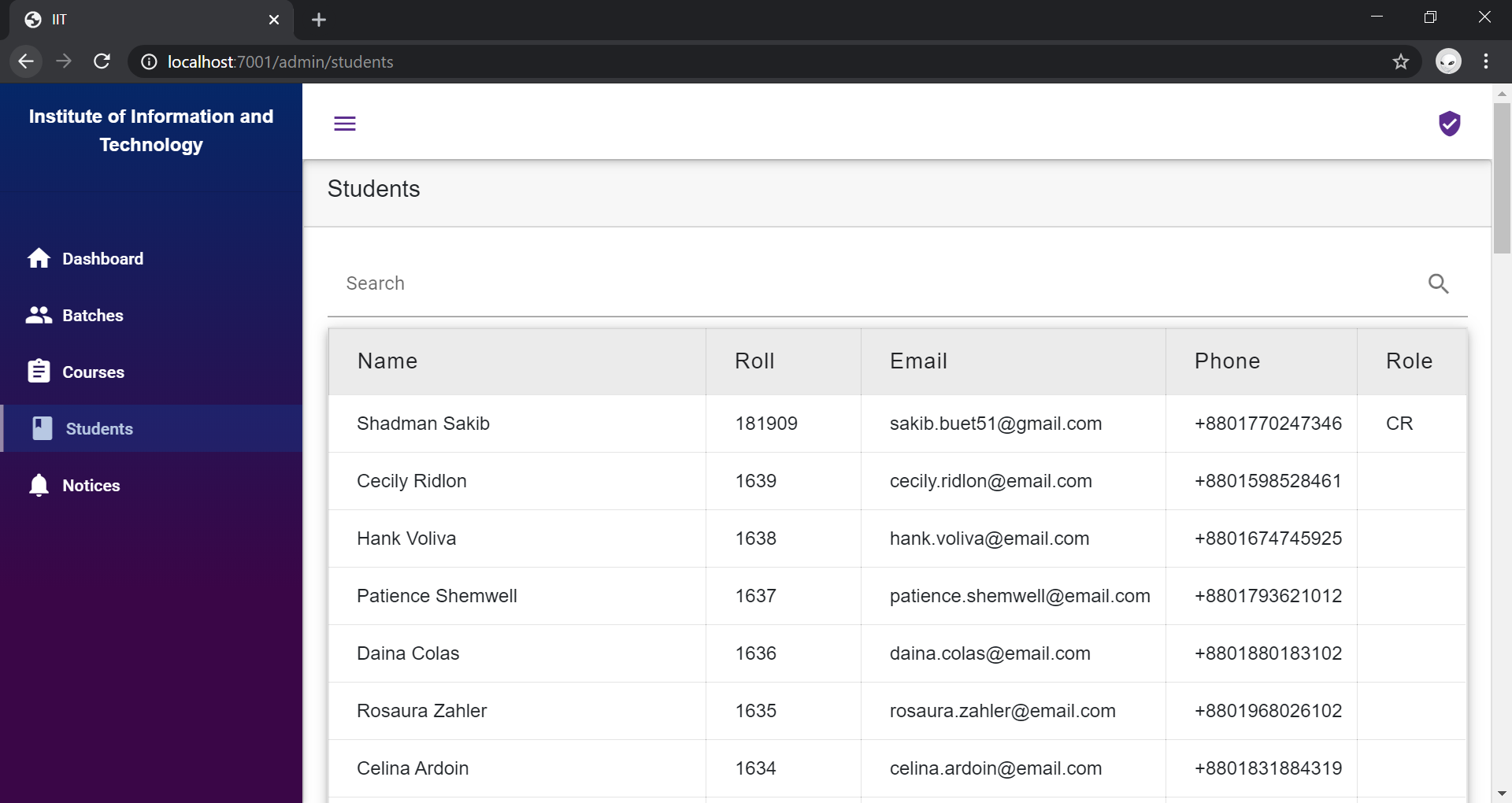
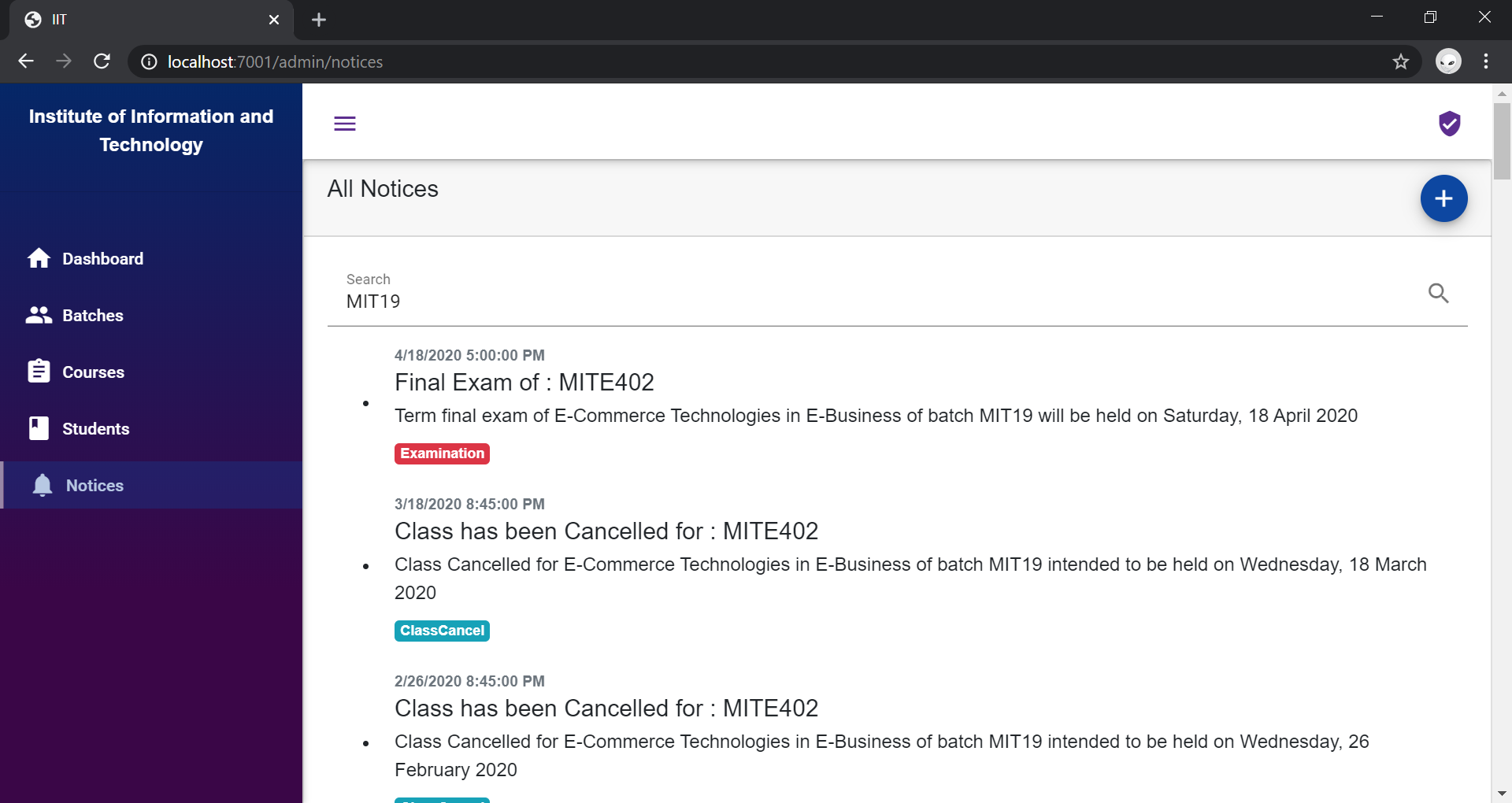


Figure 08 Admin Panel Students

## Mobile

The mobile application has been tested on an Android 9.0 running device. Here are some of the screenshots takes from the device showing the User interface of the application.

|  |  |
| --- | --- |
| C:\Users\sakib\AppData\Local\Microsoft\Windows\INetCache\Content.Word\Screenshot_20200101-163821.png  Figure 09 Mobile Student Login | C:\Users\sakib\AppData\Local\Microsoft\Windows\INetCache\Content.Word\Screenshot_20200101-163842.png  Figure 10 Mobile Dashboard |

|  |  |
| --- | --- |
| Figure 11 Mobile All Semester And Courses | Figure 12 Mobile Current Semester And Classes |

|  |  |
| --- | --- |
| Figure 13 Mobile Course Information | Figure 14 Mobile Course Create |

|  |  |
| --- | --- |
| Figure 15 Mobile Notices | Figure 16 Mobile GradeSheet |

# Conclusion

The project implements a cross platform mobile application for the students of IIT. The separate admin panel provides an easier interface also to mange course and lessons. The system takes advantage of the Mobile Push notification to communicate with students. Students can track their own academic events as well as manage custom work items.

#### Future Scope

The followings components can be implemented as a future work of the project.

* The admin panel can be fully integrated with the current IIT,DU admin panel
* The web panel for the student can be developed using existing features.
* Visual User interface can be improved.
* Social media integration can be implemented.
* Unit Testing of possible functions.

Though related, the following components are not a part of the project scope.

* The project does not intend to implement a common web user interface.
* Payment Gateway integration is out of scope
* Admission System requires external modules.

# References

1. **Microsoft.** visualstudio. *https:www.visualstudio.com.* [Online]

2. **Xamrin.** Xamarin. *https://dotnet.microsoft.com/apps/xamarin.* [Online]

3. **Microsoft.** Enterprise Application Patterns using Xamarin.Forms. *https://docs.microsoft.com/en-us/xamarin/xamarin-forms/enterprise-application-patterns/.* [Online]