PUNE VIDYARTHI GRIHA’S

COLLEGE OF ENGINERRING NASHIK



**MINI PROJECT REPORT ON**

**“ STUDENT MANAGEMENT SYSTEM”**

# FOR THE COURSE

THIRD YEAR BACHELORS OF ENGINERRING IN

INFORMATION TECHNOLOGY

**SUBMITTED BY**

Master. Pravin Chaudhari

Master. Meghana Chaudhari

Master. Apeksha Ahire

Master. Mohit Chavan

**GUIDED BY**

Prof. S.K.Thakare

**SUBMITTED TO**

PUNE VIDYARTHI GRIHA'S

## COLLEGE OF ENGINEERING NASHIK

PUNE VIDYARTHI GRIHA’S

COLLEGE OF ENGINERRING NASHIK



# CERTIFICATE

This is to certify that, the project Report on **" Student Management System "** has been successfully completed by project team in the fulfillment of requirement of Third year B. E. in **"Information Technology"** from **"Pune Vidyarthi Griha's College of Engineering Nashik"** during the Academic Year 2022-2023. It is record of their work carried in my guidance. They have satisfactorily completed this project.

## Date: 10-11-2022 Place: Nashik

**Prof. S.K.Thakare Prof. S. N. Bhadane Dr. A. R. Rasane**

**Guide HOD, IT Dept. Principal (I/c)**

# ACKOWNLEDGEMENT

We feel great in submitting this mini project report on **" Student Management System "**. We are thankful to the owner of "Balaji Book Store" for providing us with the best of the possible help for completing our project. A successful project is a result of good term work, which contains not only the people who put in their logic and hard work but also the people who guide them. We are extremely grateful for the necessary information, with support provided by **Prof. S.K.Thakare** Mam, for her timely suggestion and valuable guidance depth to her for spending her precious time with us and help us by giving the very important details regarding the project. We would also like to thank for her constant encouragement and guidance to us. Last but not least we would like to thank all the people who have helped us directly and indirectly in our project.

**Master. Pravin Chaudhari**

**Master. Meghana Chaudhari**

**Master. Apeksha Ahire**

**Master. Mohit Chavan**

**Information Technology**

(Third Year, V Semester)

# ABSTRACT

Students form a main part of any institution that concerns with. But the institutions find it difficult to keep details of so many students of the organization just in one stretch. It will involve a lot of pen paper work. Sometimes there will be some huge heap of files bundled up and kept together in some corner of the office. If you want any information regarding the particular student then it can be obtained by just entering the roll number or the name of the student to be searched. This student management system will make the work of storing the data in an organized way.

The student management system application will help in managing the student's reports, results and exams will become easier with one such system. It will also help in saving time and effort. The user interface must be user friendly and easy to understand. The information of the particular student will be obtained in just one mouse click. Some of the features that it can include are as follows:

* Student database management: The details of the students of the organizations can be stored

in the database with the use this application.

* Courses : The add multiple courses for student to enroll

# INDEX

|  |  |  |
| --- | --- | --- |
| **Sr. no.** | **Contents** | **Page no.** |
| **1** | **Introduction**  1.1 Title  1.2 Problem Statement  1.3 Need of the solution to the defined problem | 6 |
| **2** | **Software requirement and specification**  2.1 Software Requirements  2.2 Hardware Requirements  2.3 Feasibility Study | 7 |
| **3** | **System Design and architecture**  3.1 ER diagram  3.2 Database Design  3.3 Detailed Description | 9 |
| **4** | **Results/Screenshots** | 13 |
| **5** | **Other Aspects**  5.1Advantages  5.2 Disadvantages  5.3 Application | 20 |
| **6** | **Conclusion** | 22 |
| **7** | **Reference** | 23 |

## 1. INTRODUCTION

This project on “Student Management System” is useful for easy user interface. The system utilizes the powerful database management, data retrieval and data manipulation. This project provides more ease for managing the data than manually maintaining in the documents. The project is useful for saving valuable time and reducing the huge paperwork.

It will help educational Institutions like schools and colleges will keep track of their student records like personal details, contact details , marks details, etc. The Internet is rapidly becoming a part of the everyday lives of a majority of people in the world. People perform various activities on the Internet and one of them is storing their data in database where they are interested in. In these database they can post the queries and they can retrieve the required data. Obviously there is a need of Student Information System software for management of students data.

**1.1 Title**

Student Management System

### 1.2 Problem Statement

There are many departments of administration for the maintenance of college information and student databases in any institution. All these departments provide various records regarding students. Most of these track records need to maintain information about the students. This information could be the general details like student name, address, performance, attendance etc or specific information related to departments like collection of data.

.

### 1.3 Need of the solution to the defined problem

It will help educational Institutions like schools and colleges will keep track of their student records like personal details, contact details , marks details, etc. The Internet is rapidly becoming a part of the everyday lives of a majority of

people in the world

**2. SOFTWARE AND HARDWARE REQUIREMENTS**

## 1. Hardware Requirements

|  |  |  |
| --- | --- | --- |
| **Number** | **Description** | |
| 1 | PC with 250 GB or more Hard disk. | |
| 2 | PC with 2 GB RAM. | |
| 3 | PC with Pentium 1 and above. | |
|  | |  |

1. **Software Requirements**

|  |  |  |
| --- | --- | --- |
| **Number** | **Description** | **Type** |
| 1 | Operating System | Windows 11 / Windows/Linux |
| 2 | Language | Javascript, Node JS |
| 3 | Database | MongoDB |
| 4 | IDE | Visual Code |
| 5 | Browser | Google Chrome |

1. **Feasibility Study:**

This Student Managment System provide Feasibility is the test according to its work ability, Impact on the organization, ability to meet user's need and use of resources -- The objective of feasibility study is to acquire a sense of its scope.

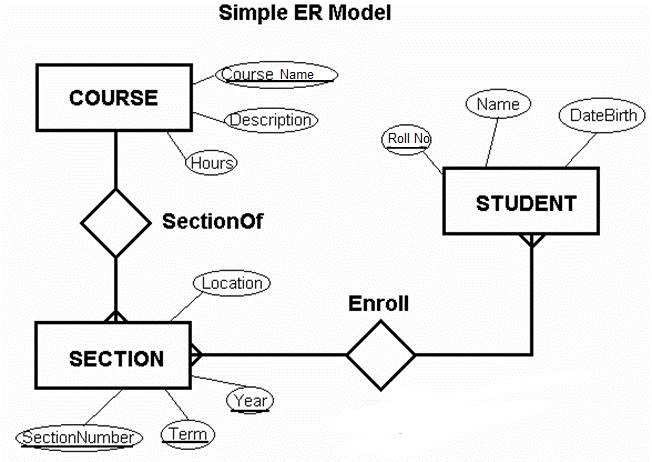
The feasibility study concentrates on 3 areas:

1. Technical feasibility
2. Educational feasibility
3. Operational feasibility

## 3. System Design and architecture

### 3.1 ER Diagram

ER Diagram stands for Entity Relationship Diagram, also known as ERD is a diagram that displays the relationship of entity sets stored in a database. In other words, ER diagrams help to explain the logical structure of databases. ER diagrams are created based on three basic concepts: entities, attributes and relationships. ER Diagrams contain different symbols that use rectangles to represent entities, ovals to define attributes and diamond shapes to represent relationships. At first look, an ER diagram looks very similar to the flowchart. However, ER Diagram includes many specialized symbols, and its meanings make this model unique. The purpose of ER Diagram is to represent the entity framework infrastructure. ER Model stands for Entity Relationship Model is a high-level conceptual data.



### 3.2 Database Design

Database is an integrated collection of data and provides a centralized access to the data and makes possible to treat data as a separate resource. Usually, centralized data managing software is called a Relational Database Management System (RDBMS). The most significant different between RDBMS and other type of Data Management is the separation of data as seen by the program and data as store of on the direct access storage device. This is the difference between logical and physical data.

## Data Dictionary

The efficiency of an application developed using MongoDB RDBMS mainly depend upon the database tables, the fields in each table and the way the tables are opened using the contents in them to retrieve the necessary information.

Hence a careful selection of tables and their fields are imperative.

The database tables used in this system are created keeping the above points in mind. The tables used are given below.

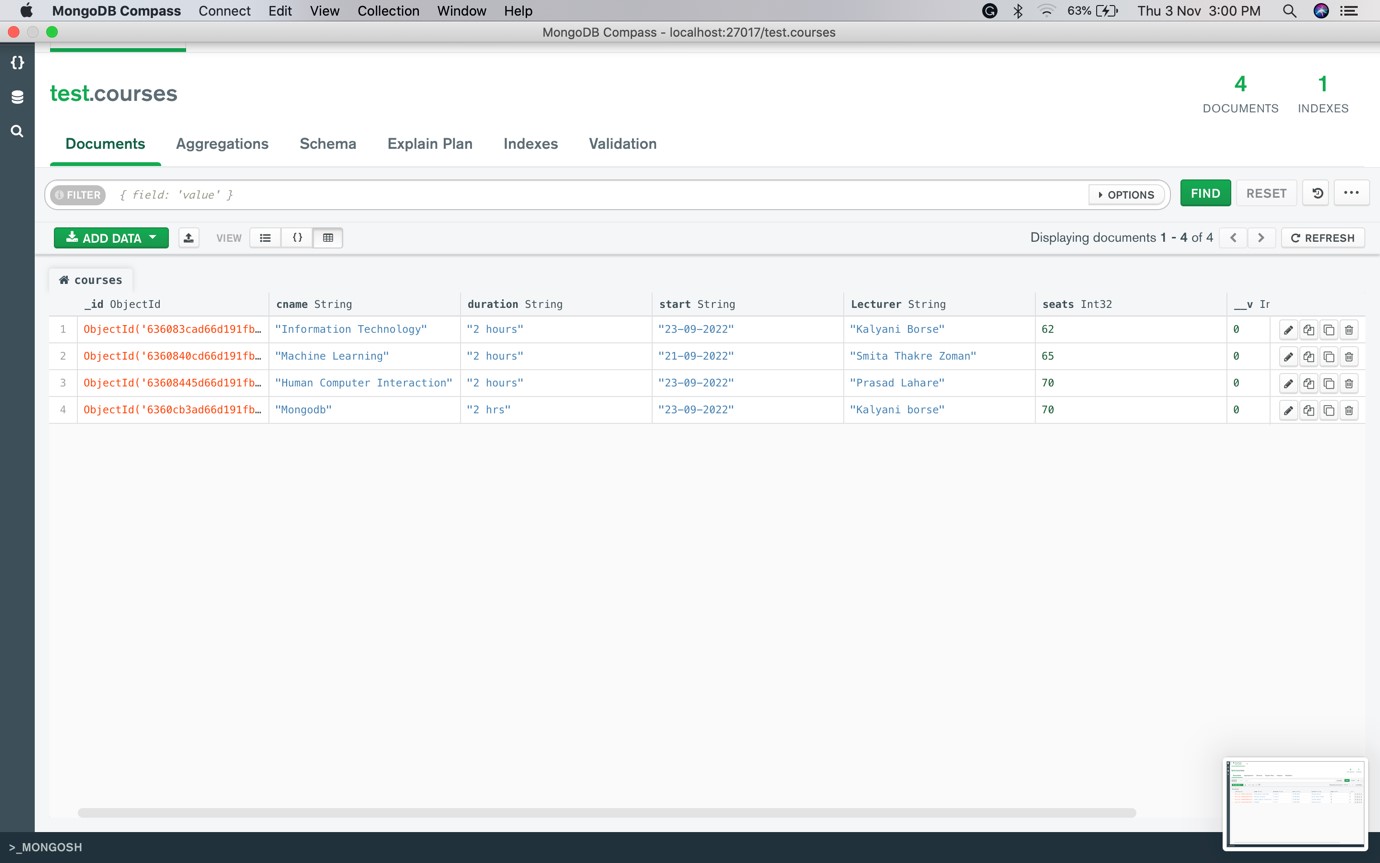
**Course collection :**

|  |  |  |
| --- | --- | --- |
| **#** | **Name** | **Type** |
| 1 | **\_id** | objectId |
| 2 | **cname** | String |
| 3 | **duration** | String |
| 4 | **Start** | String |
| 5 | **lecturer** | String |
| 6 | **seats** | Int32 |

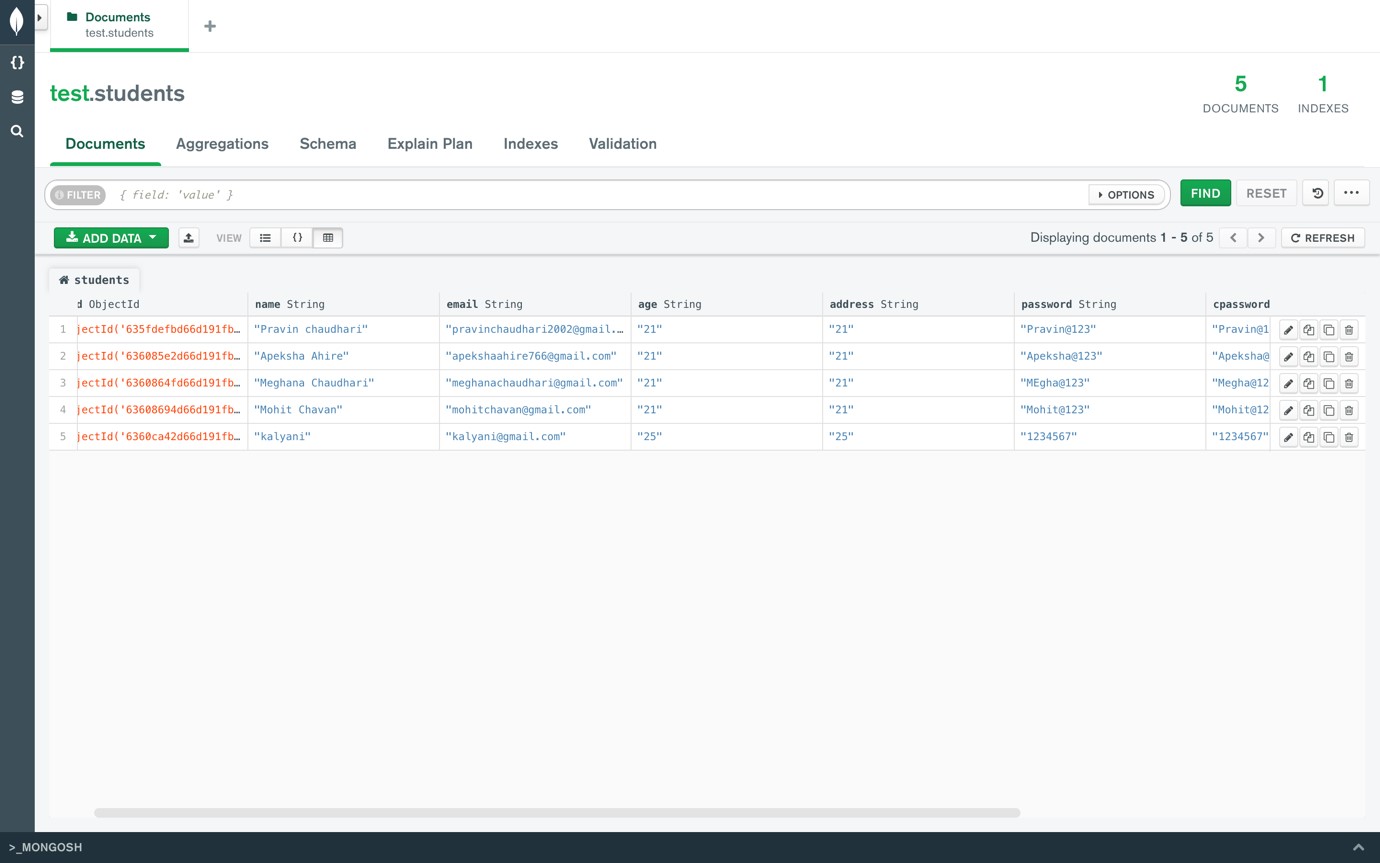
**Student collection:**

|  |  |  |
| --- | --- | --- |
| **#** | **Name** | **Type** |
| 1 | **\_id** | objectId |
| 2 | **name** | String |
| 3 | **email** | String |
| 4 | **age** | String |
| 5 | **address** | String |
| 6 | **password** | String |
| 7 | **cpassword** | String |

Course Schema:



Student schema:



### 3.3 Detailed Description

My project Student Managmnet System is to provide services for the managing the student and courses management. This project mainly elucidates the modules such as:

* AddCourse
* Student

**AddCourse**

The admin section contains add the courses the add courses actually inquire the admin to fill the essential field like course name, start, date, lecturer, time and seats after filling all this fields the data is send to the database. And this data is show to the student panel to enroll the multiple courses as per student

**Student**

This is the student section, for this student section the student must be sign up there are some field like name, email, password etc after filling this essential fields we create a account of student where student is login by email id and password that is set by user After the login there are some courses are available to enroll this courses student must have to check time course duration lecturer after seeing this detail student must be click on the enroll button after this student must fetch the seats that he wants after this the student enrolled the appropriate course

|  |
| --- |
| **Coding**  Main Connection File(app.js)  var express = require('express'); var path = require('path'); var favicon = require('serve-favicon'); var logger = require('morgan'); var cookieParser = require('cookie-parser'); var bodyParser = require('body-parser');    var mongoose = require('mongoose'); var mongoDB = 'mongodb://localhost:27017'; mongoose.connect(mongoDB, { useNewUrlParser: true, useUnifiedTopology: true, useFindAndModify: false,  });  var db = mongoose.connection;    db.on("error", () => { console.log("MongoDB connection failed...");  });  db.once("open", () => { console.log("MongoDB connection successful...");  });    var index = require('./routes/index'); var students = require('./routes/students'); var newstudent = require('./routes/student\_signup'); var student\_login = require('./routes/student\_login'); var student\_course = require('./routes/student\_course'); var addcourse = require('./routes/addcourse'); var course\_enroll = require('./routes/course\_enroll'); var app = express();    12 |

|  |
| --- |
| app.set('views', path.join(\_\_dirname, 'views')); app.set('view engine', 'jade');    app.use(logger('dev')); app.use(bodyParser.json()); app.use(bodyParser.urlencoded({ extended: false })); app.use(cookieParser()); app.use(express.static(path.join(\_\_dirname, 'public')));    app.use('/', index); app.use('/students', students); app.use('/student\_signup', newstudent); app.use('/student\_login', student\_login); app.use('/student\_course', student\_course); app.use('/addcourse', addcourse);  app.use('/course\_enroll', course\_enroll);    app.use(function(req, res, next) { var err = new Error('Cannot found the requested page!'); err.status = 404; next(err);  });  app.use(function(err, req, res, next) { res.locals.message = err.message; res.locals.error = req.app.get('env') === 'development' ?  err : {};    res.status(err.status || 500); res.render('error');  });  module.exports = app; app.listen(3000);          13 |

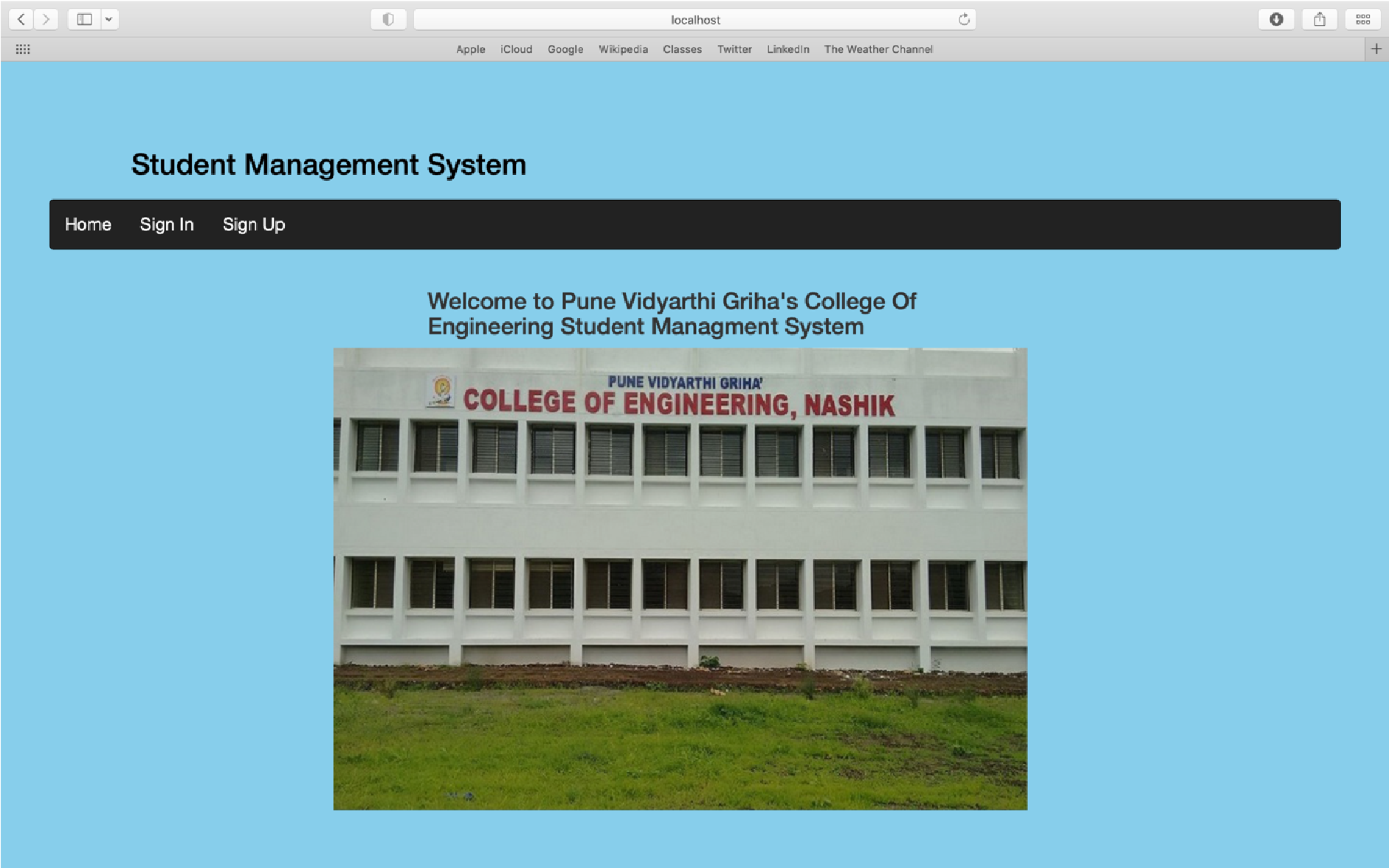
Course File(Course.js)

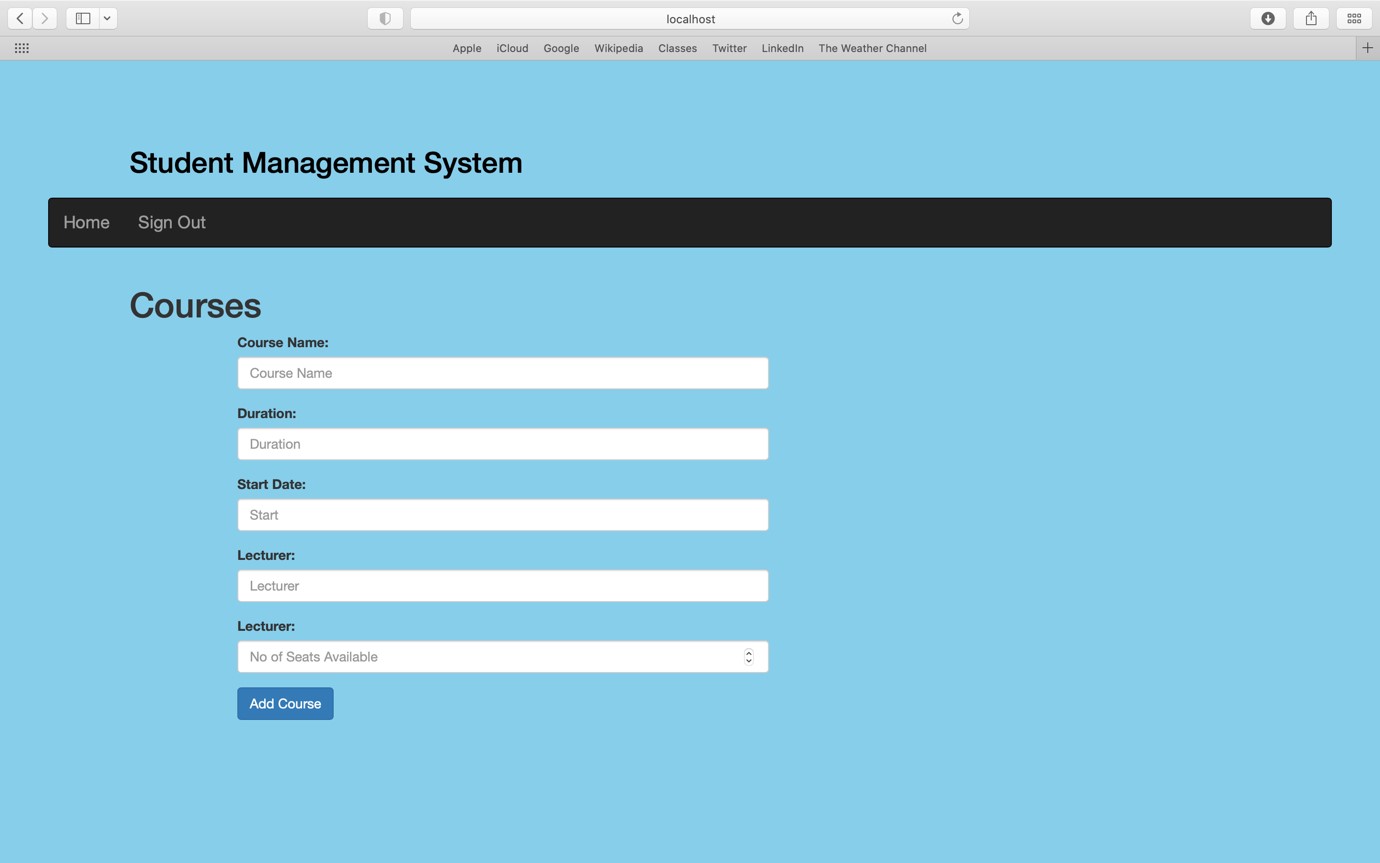
|  |
| --- |
| var mongoose = require('mongoose'); var Schema = mongoose.Schema;    var Course = new Schema({ cname: String, duration: String, start: String,  Lecturer: String, seats: Number  });  var Course = mongoose.model('Course', Course ); module.exports = { Course: Course  } |

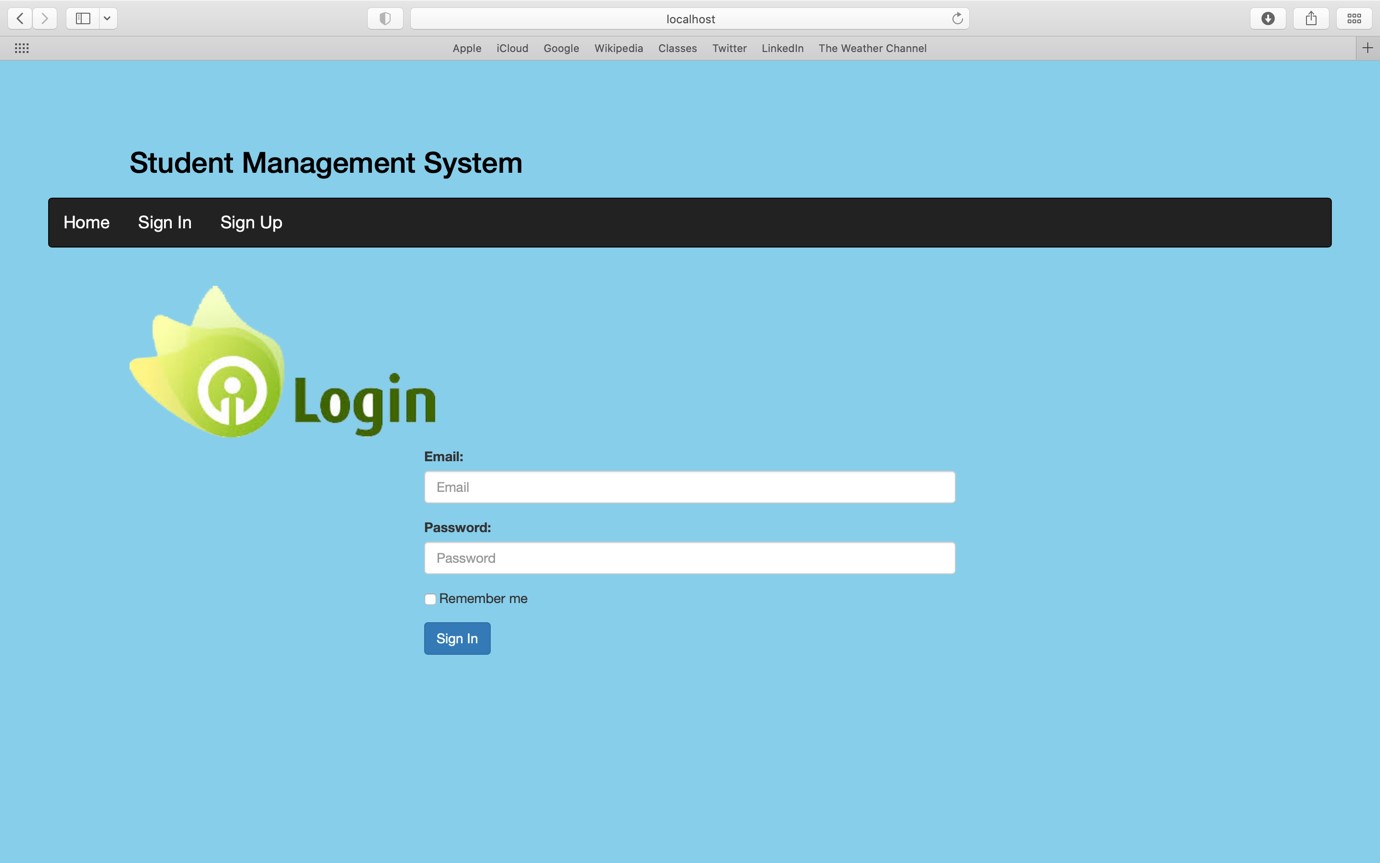
Student File(Student.js)

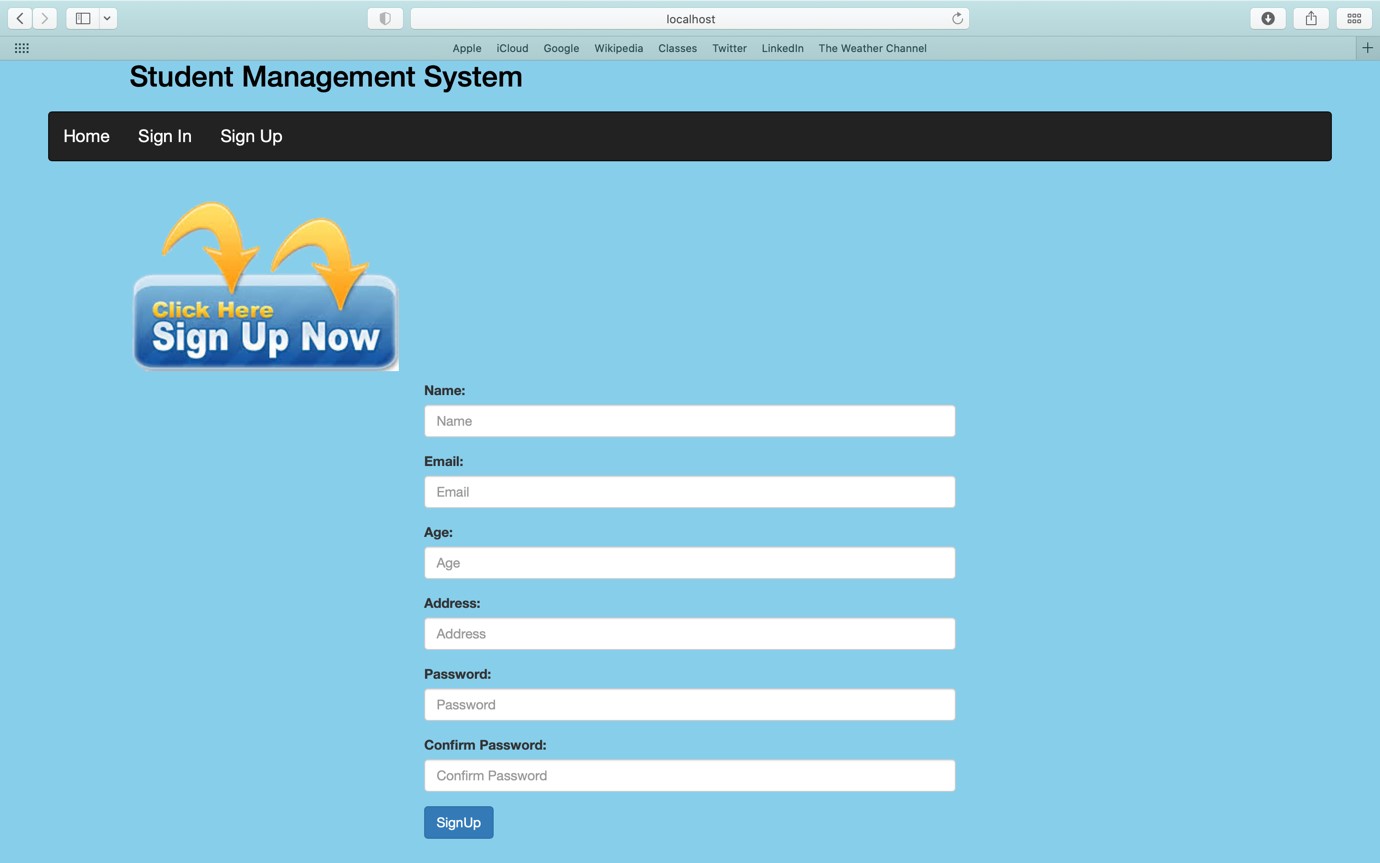
|  |
| --- |
| var mongoose = require('mongoose'); var Schema = mongoose.Schema;    var Students = new Schema({ name: String, email: String, age: String, address: String, password: String, cpassword: String  });  var Students = mongoose.model('Students', Students ); module.exports = { Students: Students  } |

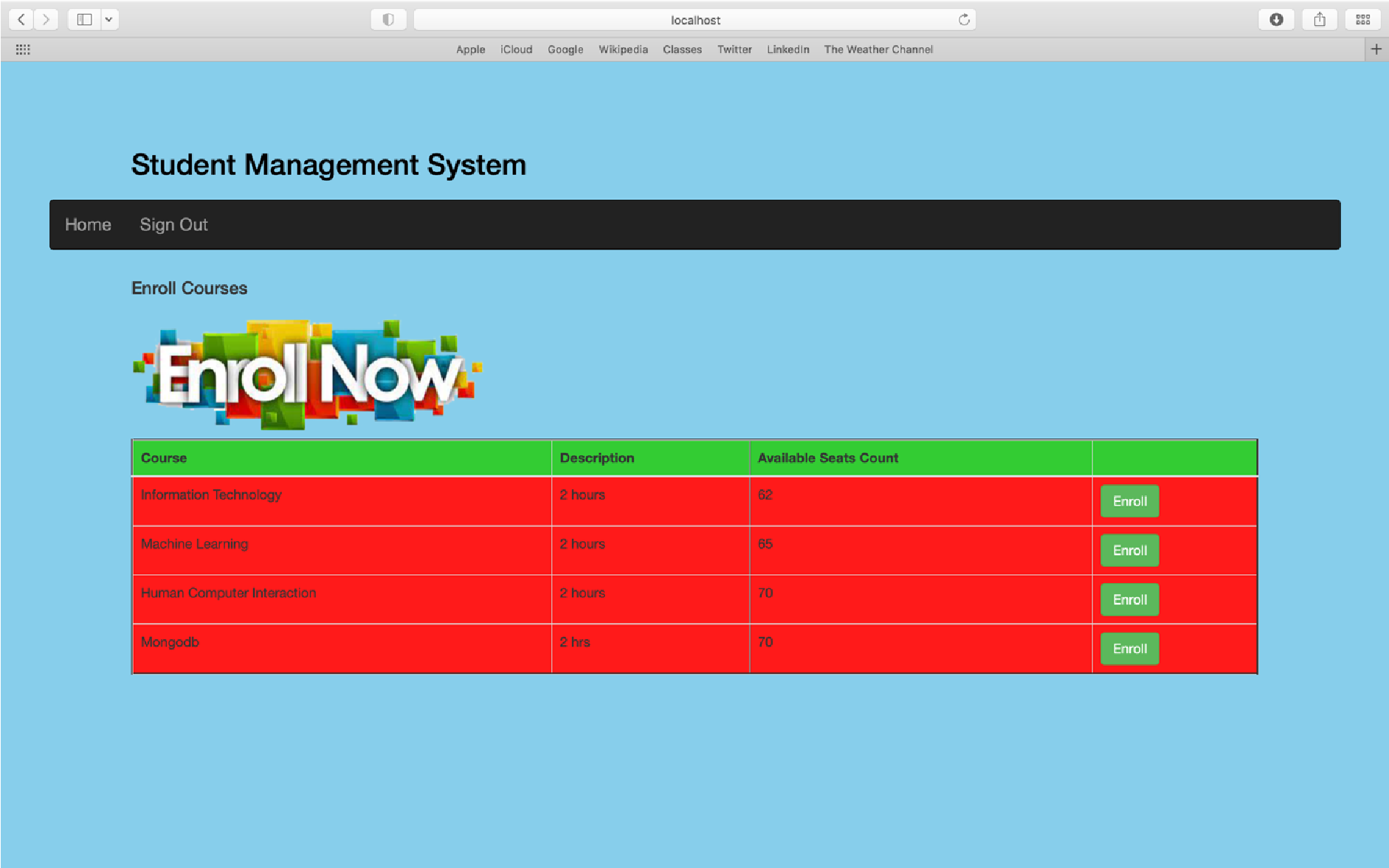
## 4. Snapshot Of Project

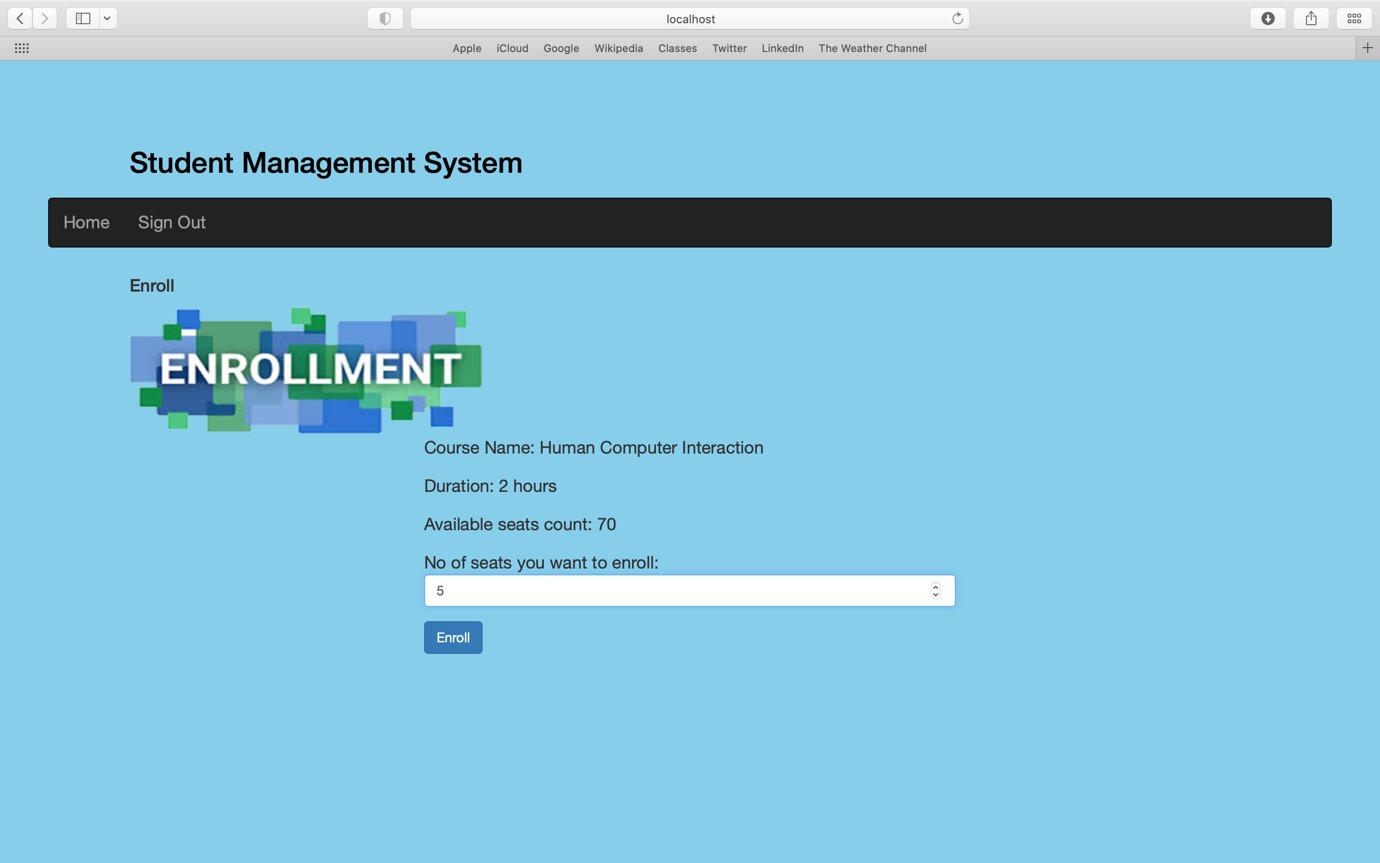


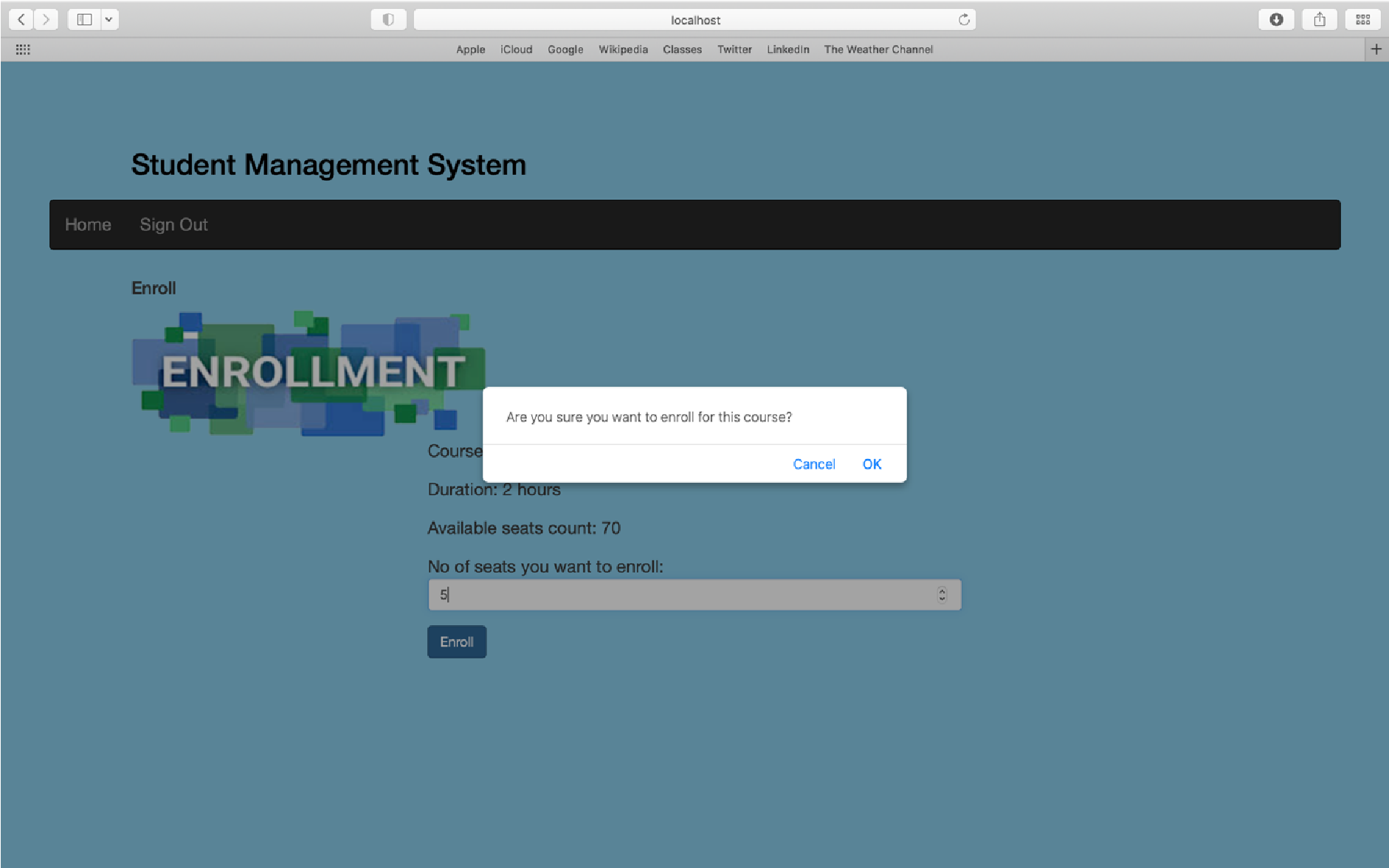












## 5. Other Aspects

**5.1 Advantages:**

* It’s user-friendly.
* Each of the actors has been provided with register id, with which he has to log on to perform the required action.
* Student’s data is highly secured in this software.
* Only the administrator has a right to register a student in database.
* Faculty or student has a right to retrieve the details of a student.
* Only teacher has the right to update the marks of a student

**5.2 Disadvantages:**

* Replication are not handled that easy n fast

**5.3 Applications:**

* Ability to manage the information of students and courses
* Support fast searching to find match of courses and student
* Data is secured
* User effective
* Effectively manage all students

## 6. Conclusion

Student management systems make faculty jobs more accessible by giving them an easy place to find and sort information. This system allows teachers and student managers to follow with their student engagement The system provides an excellent support to the corporate client because they can directly mail their problem online. A computerized surveying system provides an easy, fast access and support for the user.The usage of software increases efficiency, and decreases the effort It has been thoroughly tested and implemented

## 7. References

**BOOKS REFERRED:**

1. Pressman, Roger S., Software Engineering, fifth edition, McGrawHill Higher Education, 2001.
2. Elmasri | Navathe, Fundamentals of Database Systems, third edition, Addison-Wesley, 2000.
3. Professional ASP.NET 4.0 in C# and VB by Bill Evgen, Scott Hanselman, Devin Rader.
4. Beginning ASP.NET 4.0 in C# by Matthew MacDonald.

**WEBSITES REFERRED:**

1. www.google.com
2. www.wikipedia.com
3. www.w3schools.com
4. www.aspforums.net
5. www.forums.tutorialized.com
6. www.youtube.com