DELHI TECHNOLOGICAL UNIVERSITY

DATABASE MANAGEMENT SYSTEM PROJECT

STUDENT DATABASE MANAGEMENT SYSTEM AND PREDICTION OF RESULTS USING DATA MINING

SUBMITTED BY:

PARTICIPANT 01: SRISTI MITRA
ROLL: 2K19/CO/389 (A6)

PARTICIPANT 02: ZISHNENDU SARKER
ROLL: 2K19/CO/450 (A6)

SUBMITTED TO:
MANOJ SETHI SIR
DATABASE MANAGEMENT SYSTEM

APRIL 2021

ACKNOWLEDGEMENT:

I would like to express my deepest appreciation towards all the resources that have provided me the possibility to make progress in our report. A special gratitude I give to our Database Management System (DBMS) teacher Manoj Sethi Sir, whose stimulating suggestions and encouragement helped to coordinate in writing this project. Since saving time is one of the main priorities now-a-days, making the data record system digital and user friendly is a necessity. So, I have planned to do a case study where I showed an efficient Database System of a school. I was inspired by our subject teacher who taught us the basics of databases and SQL. According to the knowledge we gained in the class and from some online resources, I am able to finish this project paper. I am also grateful that the project enhances our knowledge towards SQL and database management structure.

ABSTRACT:

An organized and systematic office solution is essential for all universities and organizations. 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. All the modules in college administration are interdependent. They are maintained manually. So they need to be automated and centralized as, Information from one module will be needed by other modules.

With that in mind, we overhauled the existing Student Database Management System and made necessary improvement to streamline the processes. Administrators using the system will find that the process of recording and retrieving students' information and managing their classes, including marking of attendance, is now a breeze. In general, this project aims to enhance efficiency and at the same time maintain information accurateness. Later in this report, features and improvement that allow achievement to this goal will be demonstrated and highlighted.

Data Analysis can be categorized into two forms. One is used for extracting models describing important classes; another is to predict future trends. Data classification can be used to generate models which are further used to predict the unknown classes. The accuracy of the models can be examined by checking the percentage of correctly classified instances. Lot of classification algorithms are available nowadays. One of the most commonly used algorithms is decision tree because of its simplicity of implementation and easier to understand when compared to other classification algorithms. J48 is one of the effective classification methods. In this paper, the J48 algorithm is applied for analyzing a student dataset which includes academic year, department and academic grade.

CERTIFICATE:

CERTIFICATE OF APPRECIATION OF THE PROJECT

CERTIFY TO

SRISTI MITRA AND ZISHNENDU SARKER

This is to certify that "Student Database Manaagement System and Result prediction with Data Mining" embodies the original work done by Sristi Mitra and Zishnendu Sarker during this project submission as a partial fulfillment of the requirement for the Database Management System theory final project; under the supervision of Manoj Sethi sir for IV Semester, of the Delhi Technological University, Delhi.

MR. MANOJ SETHI
Database Management System

DATE

CONTENTS:

INTRODUCTION :

Storing large amounts of data is an essential concept of the modern world but storing should be feasible and user friendly to both the user and administrator. It is being used in every sector, made and maintained. An organized and systematic office solution is essential for all universities and organizations.

Background:

The Student Management System deals with all kinds of student details, academic related reports, college details, course details, curriculum, batch details and other resource related details too. It tracks all the details of a student from the day one to the end of his course which can be used for all reporting purposes, tracking of attendance, progress in the course, completed semesters years, coming semester year curriculum details, exam details, project or any other assignment details, final exam result etc.

Purpose of the project:

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. Also the information of faculty , maintenance , student result records need to be stored . Manually storing and maintaining all this data needed a huge place and manpower . Along with that , the database system does not provide the good or bad situation prediction based on the previous results . So , our purpose is to build a student database management system which will provide us with good or bad result predictions based on previous results .

Idea of the project:

Data mining is a step in Knowledge Discovery in Databases (KDD) and aims to discover useful information from huge amounts of data. The major role of data mining is applying various procedures and algorithms in order to retrieve patterns from huge data. Nowadays data can be taken from different kinds of large volumes of datasets in various formats like flat files, videos, records, texts, images, audios, scientific data and new kinds of data formats. The data collected from different sources require proper data analysis for an efficient decision making process. With that in mind, we overhauled the existing Student Database Management System and made necessary improvement to streamline the processes.

Administrators using the system will find that the process of recording and retrieving students' information and managing their classes, including marking of attendance, is now easy as a breeze. In general, this project aims to enhance efficiency and at the same time maintain information accurateness along with future result prediction. Later in this report, features and improvement that allow achievement to this goal will be demonstrated and highlighted.

PROBLEM STATEMENT :

•Problems occurred before having computerized system includes:

❏ File lost: When computerized system is not implemented file is always lost because of human environment. due to some human error there may be a loss of records.

❏ File damaged: When a computerized system is not there file is always lost due to some accident like of water by some member on file accidentally. Besides some natural disasters like floods or fires may also damage the files.

Large storage: Manually storing data needs a lot of space to store. When we computerized all the information of the students and others, it comprised a very small place rather than lots of paperwork.

❏ Result prediction: Students does not get fruitful results back from their institution because of lack of time and scope to see all the students one by one and find out their flaws. It is not always possible to predict how much a student may score from their institutional records and behaviors. So, the school system does not become beneficial for the students.

OBJECTIVE:

- Our overall aim is to, by the end of this project, be able to create a student management system. A student management system deals with all kinds of student details. Details such as student personal information, academic related reports, college or university, course details, curriculum and other student related information. All in all, this system helps the school faculty to manage data, communications, and scheduling.
- The major objective of this work is to use data mining methodologies to analyze a student's result based on their academic performance. Data mining provides many tasks that could be used to analyze the performance of the student. In this research, the

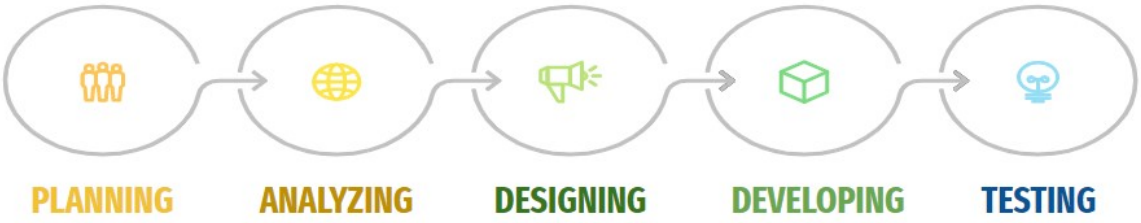
classification is used to predict the student final result based on their academic performance. As there are many algorithms used for data classification, in this paper the decision tree method is used for classification.

- The student management system is an automated version of the manual Student Management System. It can handle all details about a student. The details include college details, subject details, student personnel details, academic details, exam details etc.
- We are planning to utilize the powerful database management, data retrieval and data manipulation. We will provide more ease for managing the data than manually maintaining in the documents. Our work is useful for saving valuable time and reduces the huge paperwork. Our work is useful for easy user interface.

RELATED WORK:

System Development Life Cycle (SDLC):

Systems Development Life Cycle (SDLC) is the most common process adopted to develop a project and not surprisingly, this project is following this model too. To be precise, the waterfall model is being applied. Waterfall model is a sequential model process where the input of a phase actually results from the previous phase.



There are five phases in this model and the first phase is the planning stage. The planning stage determines the objectives of the project and whether the project should be given the green light to proceed. This is where the proposal submission comes into picture. After obtaining the approval, the next phase is analysis. Gathering and analysing the system and user requirements is essential for entry to the design step.

Data Mining:

Data mining has been widely applied in the higher education field as private arts and science colleges, Engineering Colleges, Polytechnic Colleges and universities provide huge amounts of data. Some of the application is to study features that affect student retention through monitoring academic performance and providing powerful methods to intervene as proposed by. Bassil proposed a model for a typical university information system that is based on transforming an operational database whose data are extracted from an already existing operational database. The purpose of the model generation is to help decision makers and university principles for efficient decision making. Romero and Ventura gave a survey in Applications of data mining in learning management systems and a case study class with the Moodle system. Fadl Elsid and Mirghani A. Eltahir applied the C4.8 algorithm for student databases to predict the student academic performance and a case study with faculty of computer science and information technology, Nile Valley University. Ahmed et al. Used the classification process to predict the final grade, by presenting an analysis which can help the student's instructors to improve the student's performance.

Scripting language selection:

There are many scripting languages available in the market. VBScript, Perl, JSP (Java Server Pages), ASP (Active Server Pages) and PHP (Hypertext Preprocessor) are some of those commonly used. Yet for this project, PHP is the language that is utilised for the coding piece because it is a server-side, embeddable HTML language. Being a widely-used open source scripting language, it is free for everyone to use and is especially suited for web development. On top of that, the existing system is already using PHP. PHP is the preferred selection due to the ease of usage and it can be uploaded and run on another platform with minimal change required to be done to the script. Beyond and above, the compiling time and speed for PHP is faster and more efficient. Other than being a freeware, there are many free upgrade packages easily available. The other benefit of choosing PHP is the ease in installation. It can run as a plug-in on quite a number of web servers such as the Apache. On the other hand, JSP requires a J2EE server to run and because it is a Java coded language, it is therefore more complex to understand and to do coding .

Database selection:

There are a variety of databases that we can select from the market. The widely used databases are Microsoft Access, Microsoft SQL, Oracle and MySQL. MySQL database is easy to install, user friendly, reliable and is able to run on different platforms. Moreover PHP can access MySQL databases directly without the need to go through ODBC (Open Database Connectivity). To conclude, PHP script is able to run faster with MySQL database and the processing time will definitely be shorter. The pre-school does not require complex and costly software for its database management system hence MySQL is the ideal database for this project.

Web server selection :

After deciding on the scripting language and database, next is to select the web server that can support them. Web server is necessary for the delivery of web content to the web browser. As such, Apache HTTP server which has performance similar to other 'high-performance' servers is considered. Thereafter, research and actual testing have been performed to see the outcome of the various servers listed in the Figure below. These servers include PHP and MySQL in their installation packages thus allowing smoother and simpler download processes. However, based on the performance and interface, XAMPP server is the preferred choice.

Software and Hardware requirements:

It needs Most recent version of Google Chrome,
Firefox, Internet Explorer, or safari.

Hardware Requirement: –

1. CPU: Single Core 2.4 GHZ RAM: 512 MB
2. Graphics Card: Intel or Nvidia
3. Hard Drive: 5 Gigabytes
4. Network: Broadband Recommended –
5. Processor: Pentium

Operating System: Window(XP, Vista, 7, 8, 10),
Mac OS, Linux, Unix.
Internet Connection with good speed.

PROPOSED WORK:

ER Diagram:

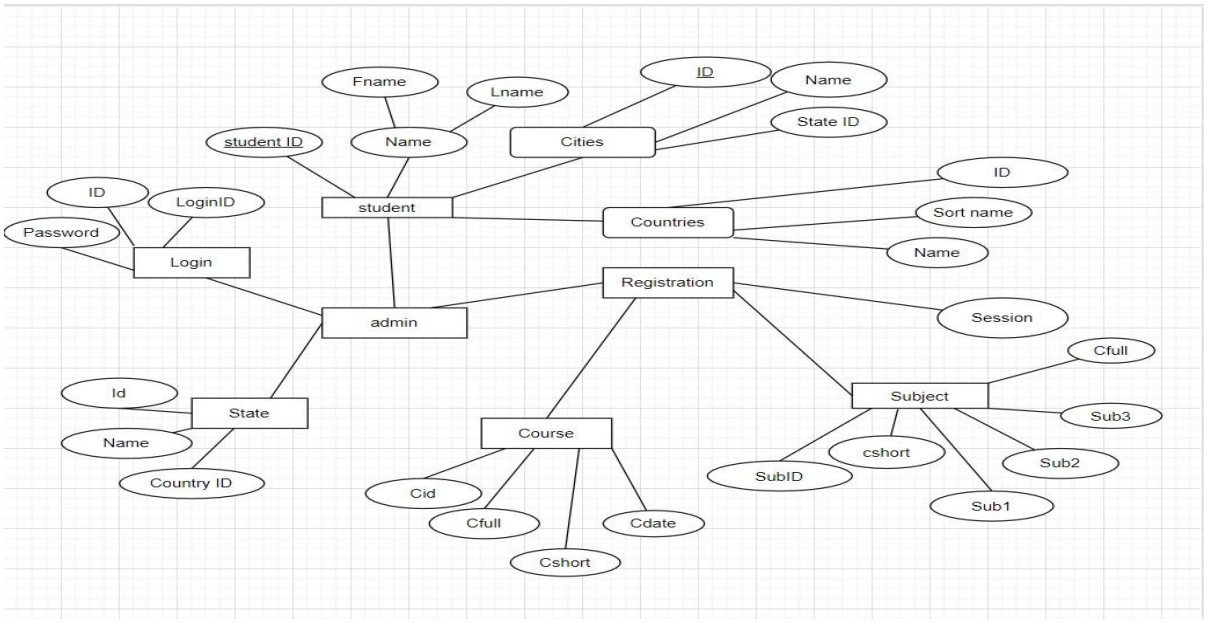


Fig : ER Diagram

Data Flow Diagram:

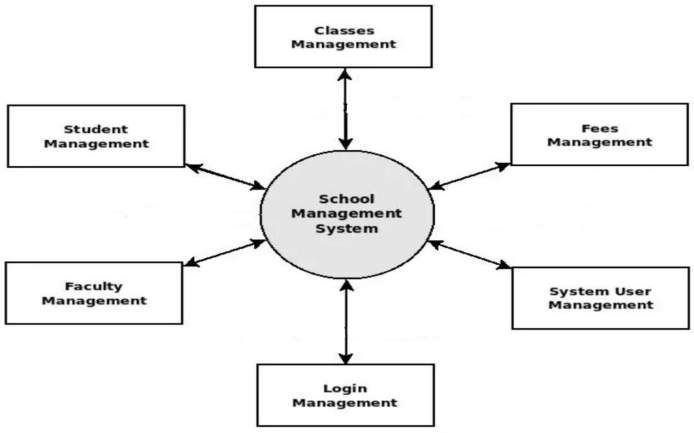


Fig: Zero level DFD

Concepts Used:

Data mining is the process of analyzing data from different angles and summarizing results into useful information. Data mining software is analytical tools which allow the users to analyze data from many different dimensions, categorize the data, and summarize the relationships among data. Technically, data mining is the process of finding correlations among many numbers of fields in a huge dataset. Five major elements of data mining are:

- Select, Transform and store data onto the data warehouse system.
- Keep and handle the data in a multidimensional database system.
- Allow business analysts and information technology professionals to access the data
- Use application software to analyze the data
- Display the data in a human understandable format, such as a graph or table.

Classification:

Classification is a data mining technique that assigns items in a collection to target categories. The objective of classification is to accurately predict the category which is unknown for each case in the data. For example, a classification model could be used to identify student results as pass, good, very good or excellent.A classification task begins with a data set with known class labels. For example, a classification model which predicts student results might be developed based on observed data for students academic performance over a period of time. In addition, the data might track previous performance,attendance percentage, general and technical attitude, and so on. Classification algorithm can be applied for categorical data. When the target is numerical the predictive model uses a regression algorithm. Comparing the values of the predictors and the values of the target gives the accuracy of the classification model. The methods for obtaining relationships can be different from one classification algorithm to another. If the accuracy percentage is acceptable, the model can then be applied to a different data set in which the class assignments are unknown.The dataset for a classification algorithm is divided into two data sets: 1.Training set is the one for building the model 2. Test set is the one for testing the model. Classification algorithms can be applied to many applications such as biomedical and drug response modeling, customer segmentation, business modeling, marketing and credit analysis. Accuracy of the model refers to the percentage of correctly classified instances made by the model when compared with the actual classifications in the test data.

- **Data Mining Process:** Data mining process discovers patterns from huge datasets involving techniques which combine artificial intelligence techniques, machine learning, statistics, Predictive analytics, and database systems. Data mining is the process of discovering useful, hidden information from a huge dataset. Data mining applies mathematical analysis to derive patterns and trends that exist in huge data. Data mining process is a series of steps and the important steps can be summarized in the following:

- **Data collection**: Data collection is the process of gathering information usually with software. The first step of the data mining process is collecting data from different kinds of sources.
- **Data Preprocessing**: Data preprocessing is a data mining technique that transforms original data into an understandable format. The real-world data is often incomplete, inconsistent, and is likely to contain many errors. Data preprocessing removes such problems. Data preprocessing prepares collected data for further processing.
- **Pattern Discovery**: Pattern is discovered by classifying students according to their graduation degrees, academic year and performance. The goal of classification is to identify the distinguished characteristics of predefined classes, based on a set of instances, e.g. students performance, academic year and degrees etc., Pattern discovery requires mining and selection of attributes which describes its properties of a given class or category.

J48 Algorithm:

The J48 algorithm is used to classify different applications and perform accurate results of the classification. J48 algorithm is one of the best machine learning algorithms to examine the data categorically and continuously. Quinlan's C4.5 algorithm actualizes J48 to create a trimmed C4.5 decision tree. The every aspect of the information is split into minor subsets to base on a decision. J48 looks at the standardized data gain that really results in the splitting of the information by choosing an attribute. To summarize, the attribute extreme standardized data gained is utilized. The minor subsets are returned by the algorithm. The split strategies stop if a

subset has a place with a similar class in all the instances. J48 develops a decision node utilizing the expected estimations of the class. J48 decision tree can deal with particular characteristics, lost or missing attribute estimations of the data and varying attribute costs. Here accuracy can be expanded by pruning .

The Algorithm

Stage 1: The leaf is labeled with a similar class if the instances belong to a similar class.

Stage 2: For each attribute, the potential data will be figured and the gain in the data will be taken from the test on the attribute.

Stage 3: Finally the best attribute will be chosen depending upon the current selection parameter.

The J48 algorithm is used to implement the Univariate Decision Tree approach, while its results are discussed. Here , we used the J48 algorithm through **weka** software to implement and analysis for the result comes from a dataset of a student database management system . It is a model for student dbms where the student, admin , institute can predict the situation of their future result using the J.48 algorithm based on their current values of entity and attributes. In a student dbms these values are taken from the academic year , academic result , course/session/for a particular time . In our project we can't be able to add the J.48 algorithm directly to our student dbms yet but we are working on it. So , as a model of that, we use **weka** software for implementing and analysing the dataset and provide predictions. Briefs of the implementation and data analysing are in next parts of the paper.

➔ **WEKA** - an open source software provides tools for data preprocessing, implementation of several Machine Learning algorithms, and visualization tools so that we can develop machine learning techniques and apply them to real-world data mining problems ➔ Weka allow the

generation of the visual version of the decision tree for the J48 algorithm. So, from the "Classifier" section select "trees" > "J4.8".

➔ Decision rules will be found based on entropy and information gain ratio pair of each feature. In each level of decision tree, the feature having the maximum gain ratio will be the decision rule

J48 ALGORITHM PROCEDURAL THEORY IN WEKA:

● The dataset for a classification algorithm is divided into two data sets: 1.Training set is the one for building the model 2. Test set is the one for testing the model

● Model is built using the training set using the J48 algorithm. The objective of the work is to apply the model for the test dataset and to find the performance of the model by classifying the new instance. This model is generated for obtaining the student job position based on their academic performance. If the performance of the model is acceptable this model may be used for the forthcoming graduates to analyze the placement status.

● The model to classify the instances of the sample training set which is provided in format is ARFF (studentdata.arff)

● WEKA source file which will then include academic year, department etc.

● The attribute section is identical to the training data.

● Once the model is generated, it can be used to classify new instances. Here classification is used to classify new instance using J48 algorithm which is implemented in WEKA by the classifier

● The classification model generation process using studentdata.arff in WEKA

● The running information of model generation is generated

● For which The graphical versions of the decision tree are appeared

● Though there are a lot of attributes in student dbms but For this analysis the relevant attributes such as academic year, department of the graduate, final academic results and job-position of the graduate only used

● Once the model is generated, the details of the model such as total number of instance in the dataset, the number of correctly classified instance, incorrectly classified instance, and accuracy of the model will be displayed

IMPLEMENTATION:

At first, the total database work needs to be stored inside the folder of XAMPP software in the htdocs folder. Then we need to open XAMPP control panel and then start apache and MySQL as given below ,

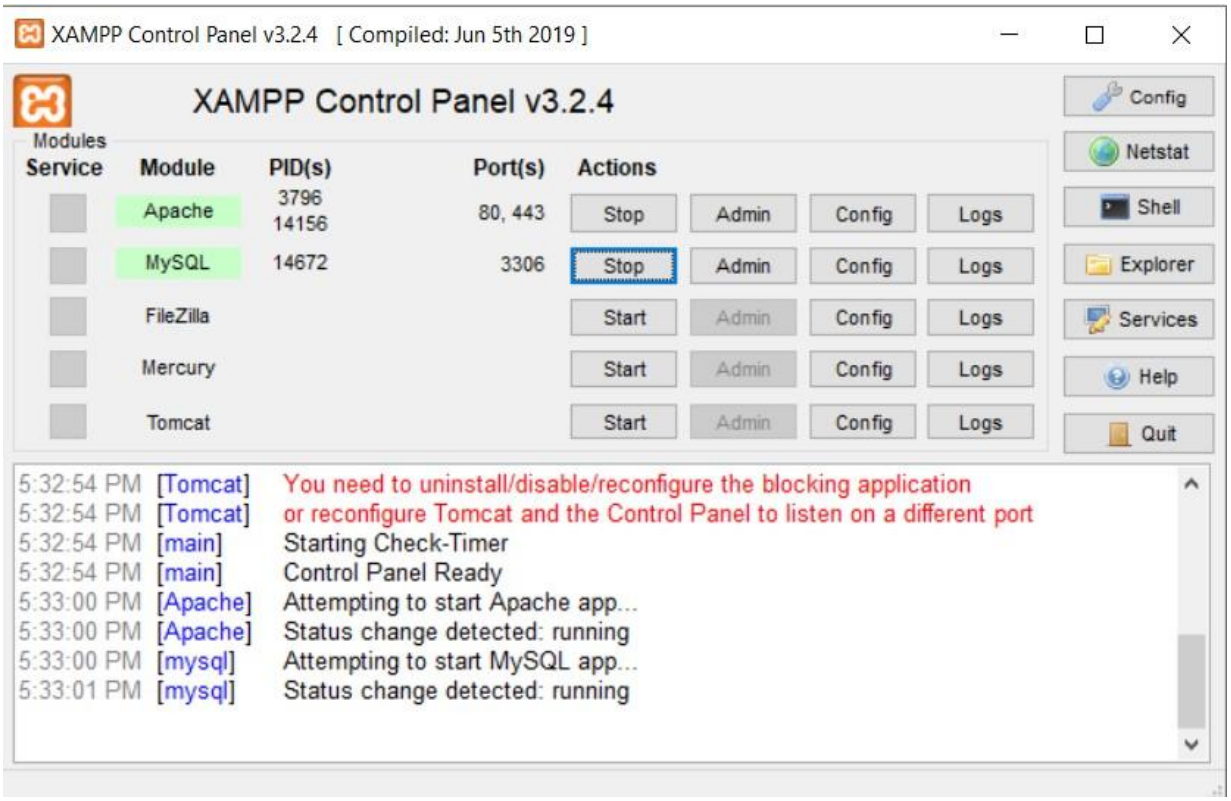


Fig: XAMPP server

After that, we need to open a browser and search for http://localhost/std_php ,then the front page of our school management system will appear and the login page will be shown , where we need to provide a proper ID and Password for login. In this case, the id is “zishnu” and the password is “sristi”.

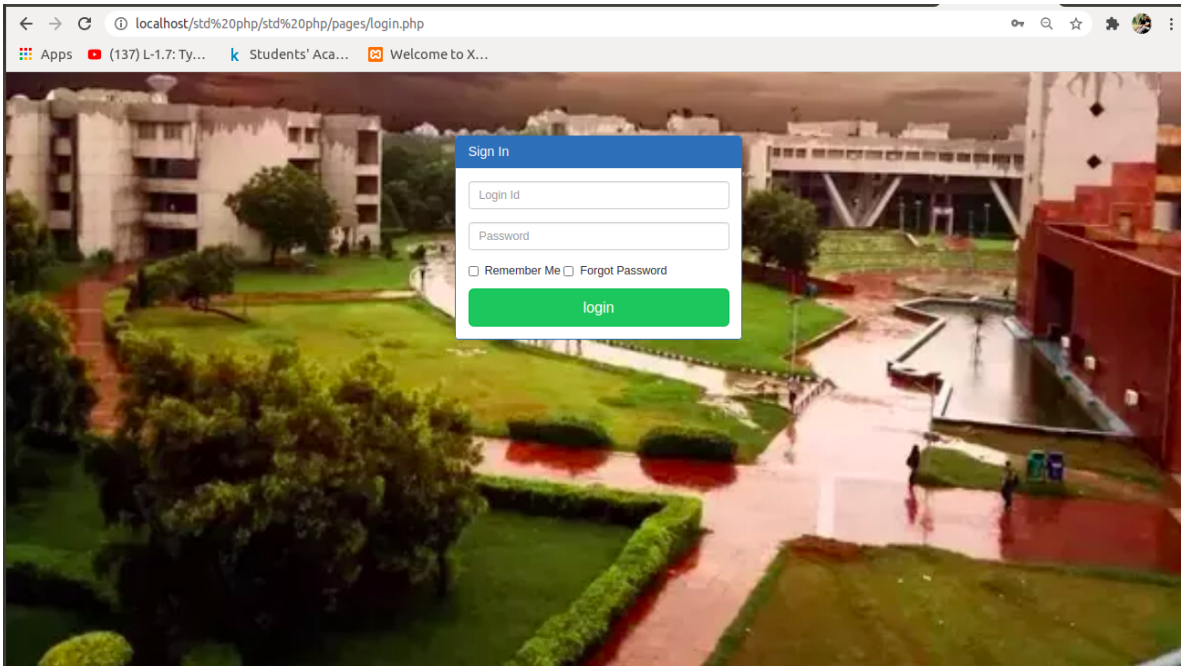


Fig : login page

Here, we are observing the admin part where we can add courses.

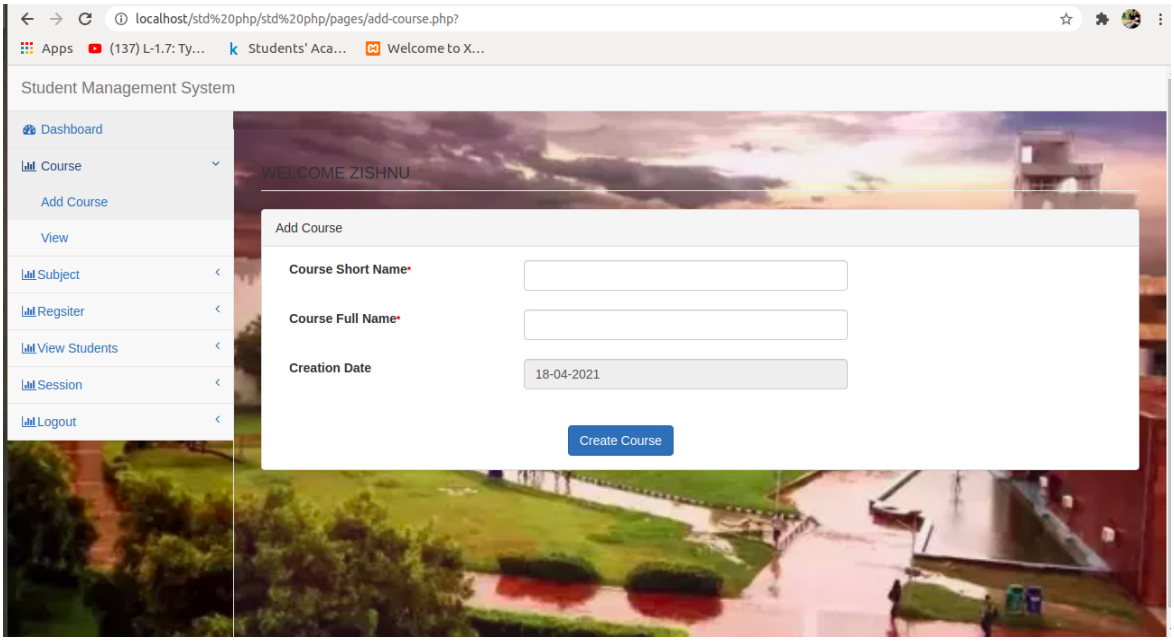


Fig : admin can add courses

We can also view the courses we added. It also shows the credited date, short name of the course and the full name of the course as well.

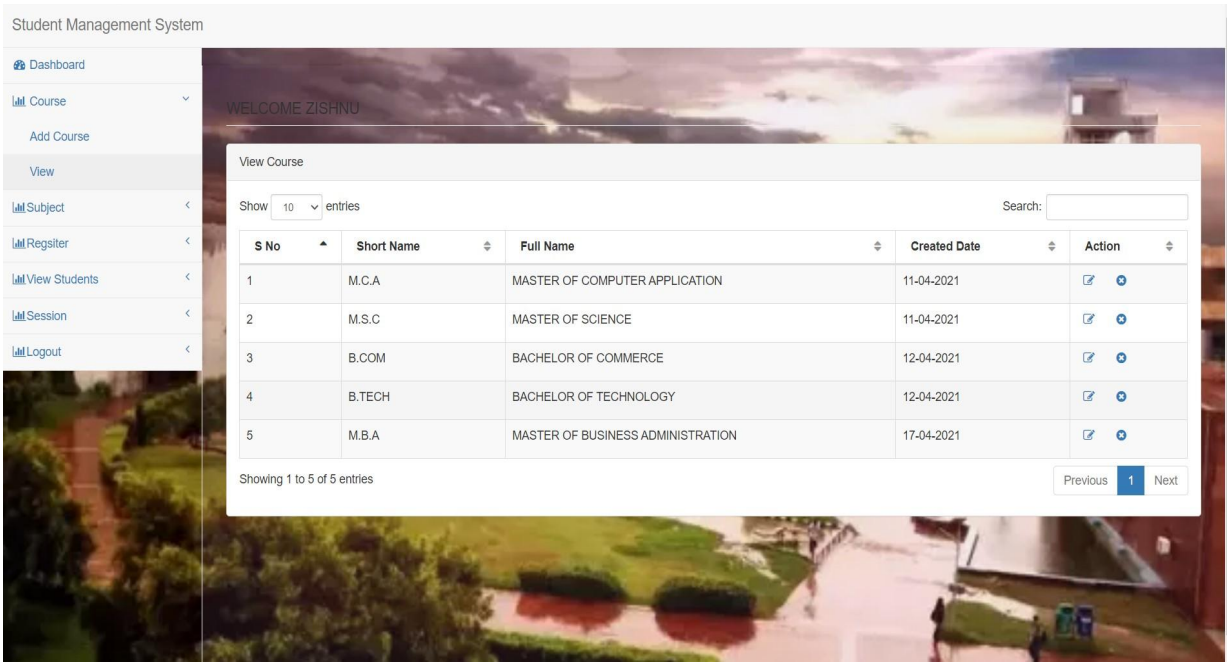


Fig: courses that are already registered

Here, we can add subjects under the courses, provided by 3 different subject names

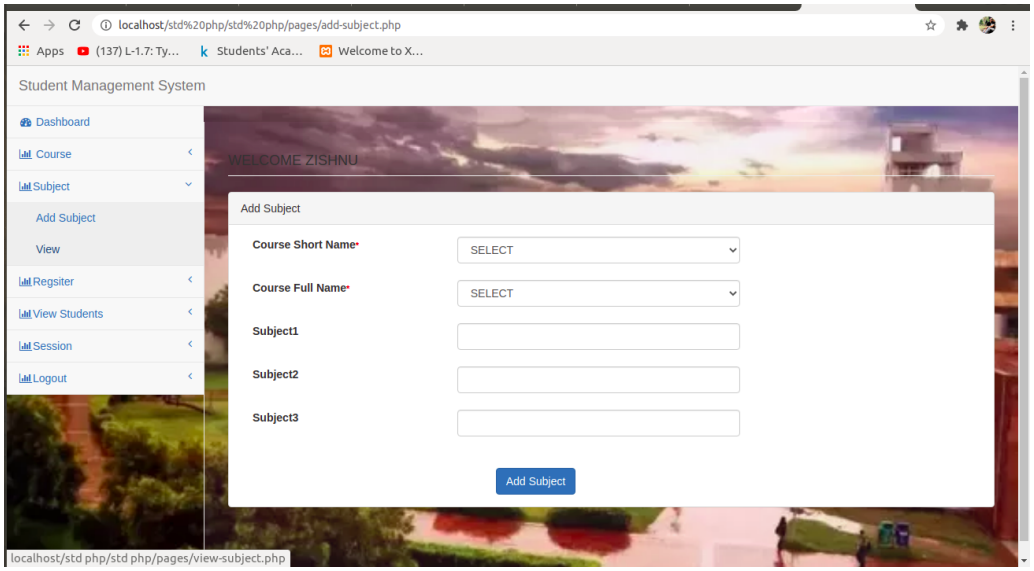


Fig : subjects will add according to course

We can view the subjects under each course and also the number of subjects added consecutively.

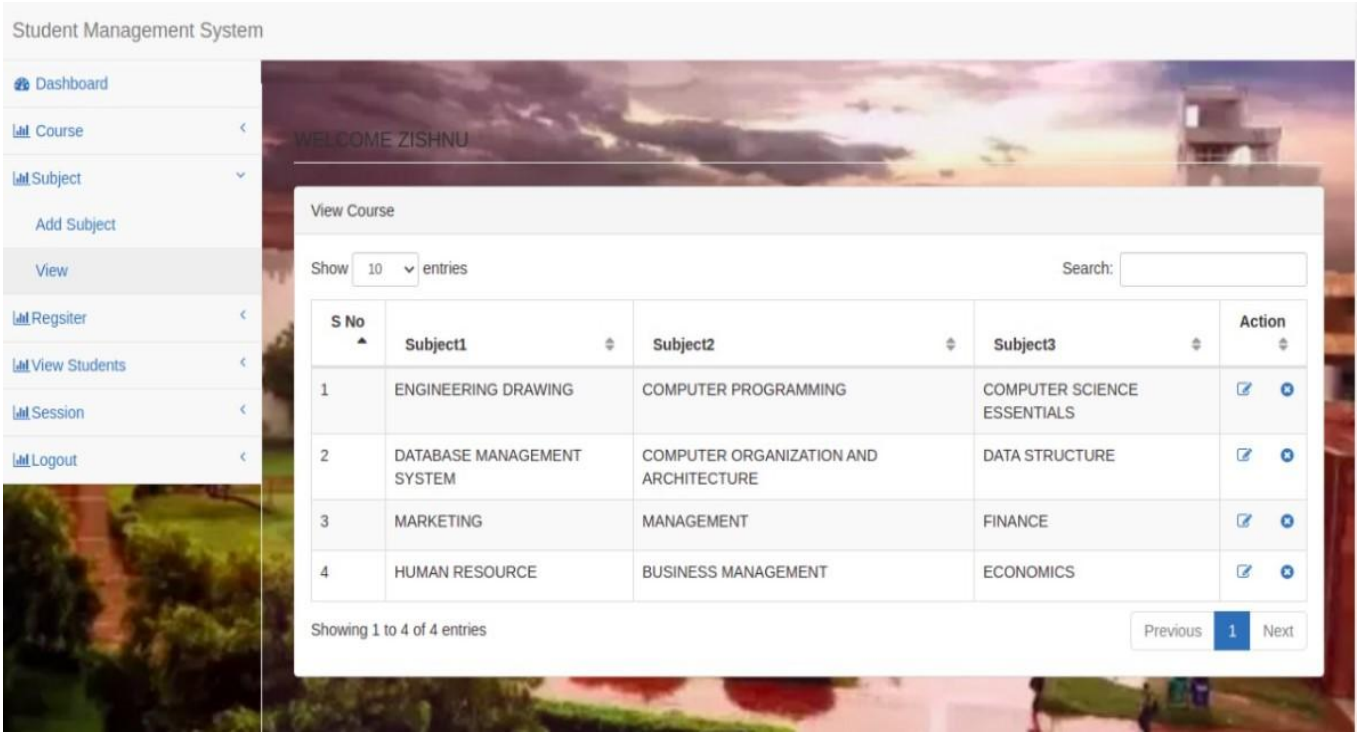
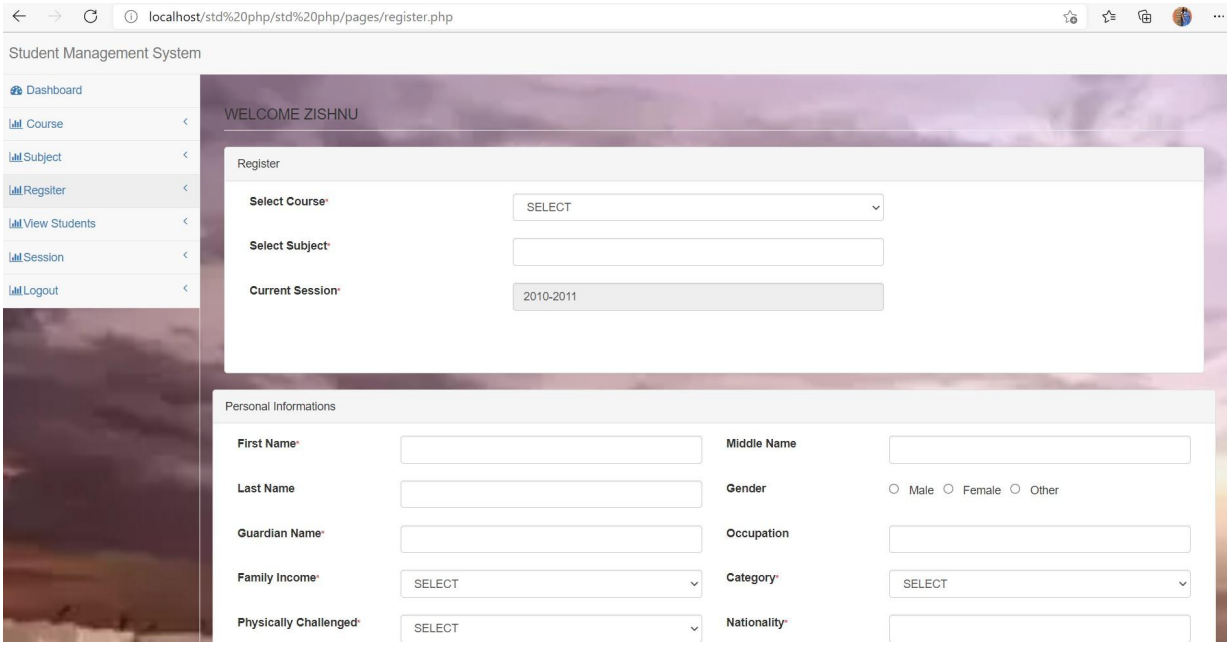
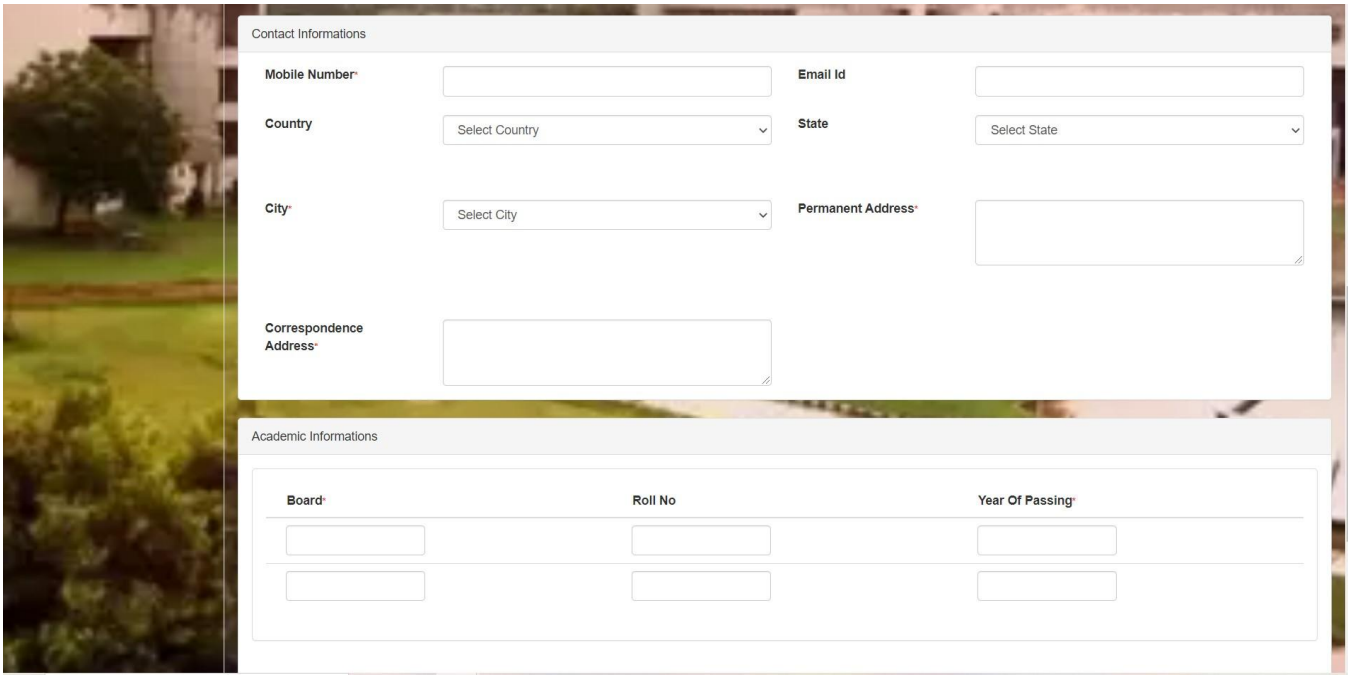


Fig : subjects that are registered

Under the existing student management module, the administrator has to first register the student. This includes the student details along with their grade. After submitting these data, the administrator will be directed to the student page whereby the student information is to be submitted.


localhost/std%20php/std%20php/pages/register.php
Student Management System
Dashboard
Course
Subject
Regsiter
View Students
Session
Logout
WELCOME ZISHNU
Register
Select Course*
SELECT
Select Subject*
Current Session*
2010-2011
Personal Informations
First Name*
Middle Name
Last Name
Gender
Male
Female
Other
Guardian Name*
Occupation
Family Income*
SELECT
Category*
SELECT
Physically Challenged*
SELECT
Nationality*


Contact Informations
Mobile Number*
Email Id
Country
Select Country
State
Select State
City*
Select City
Permanent Address*
Correspondence Address*
Academic Informations
Board*
Roll No
Year Of Passing*

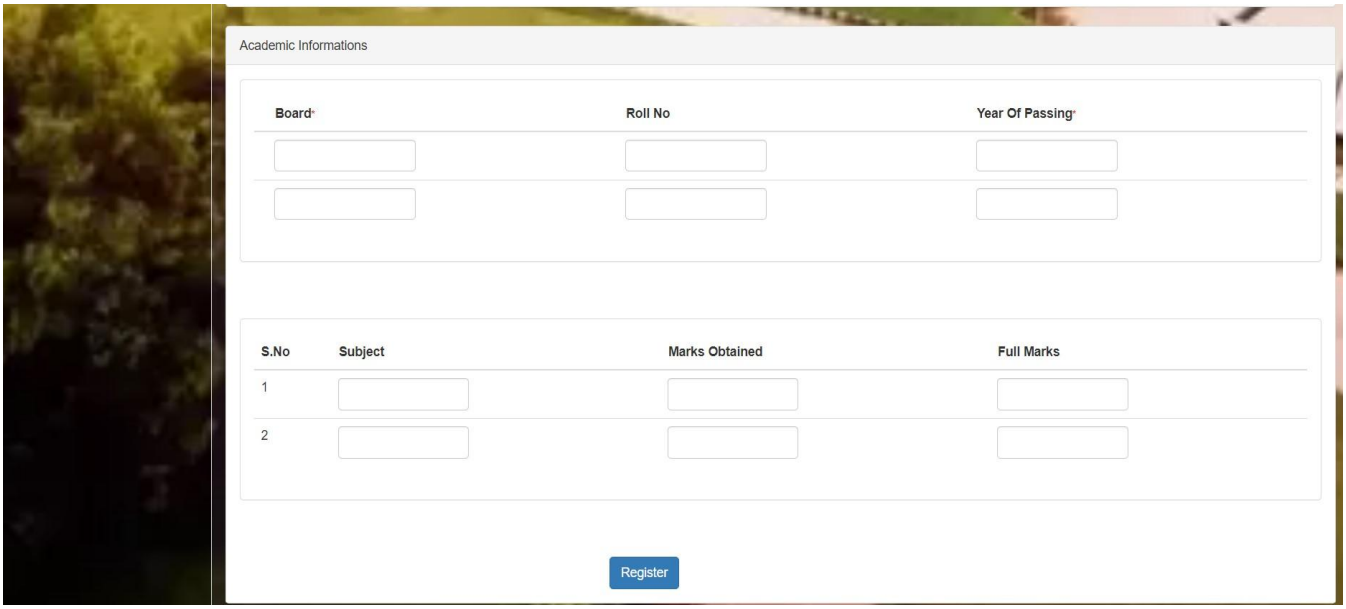


Fig : above 3 pictures are for student registration (these attributes need to register a student)

.

Here we can see the total student list and we can also sort it by date , alphabet. And also with the student name it will show some basic student details like course , subject , email mobile number and registration number.

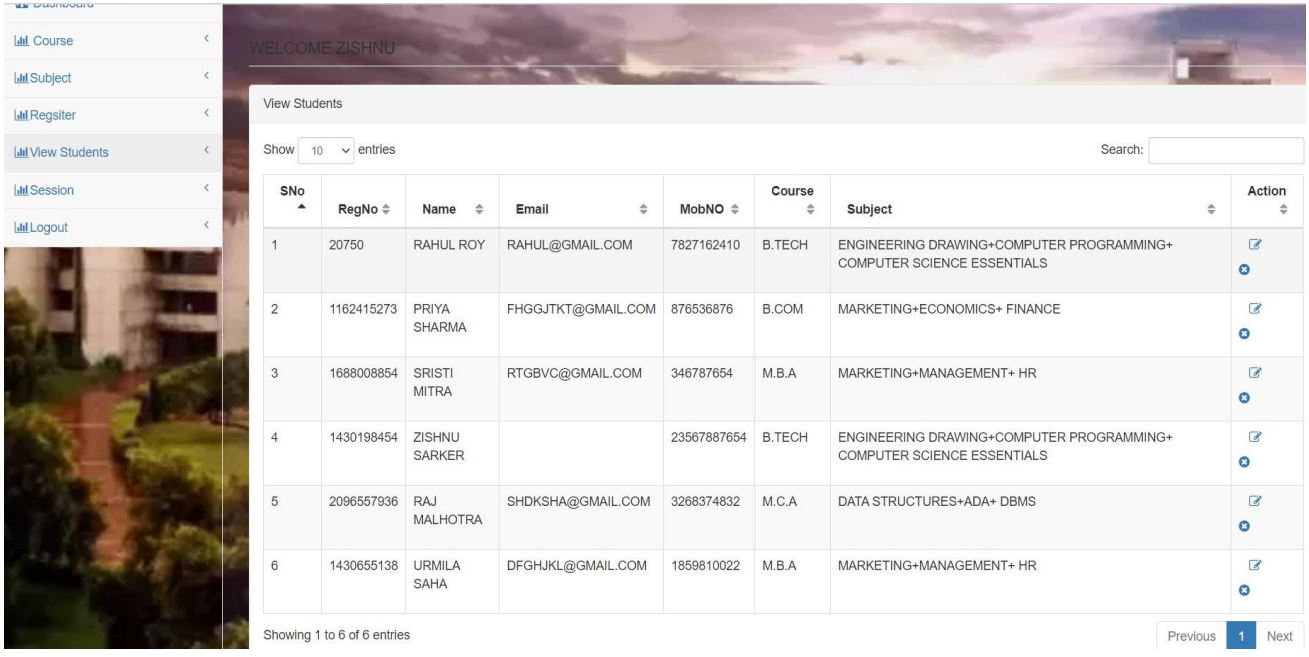


Fig : some students who are registered

At last we can search the curriculum, information and activities according to time and year by choosing a session.

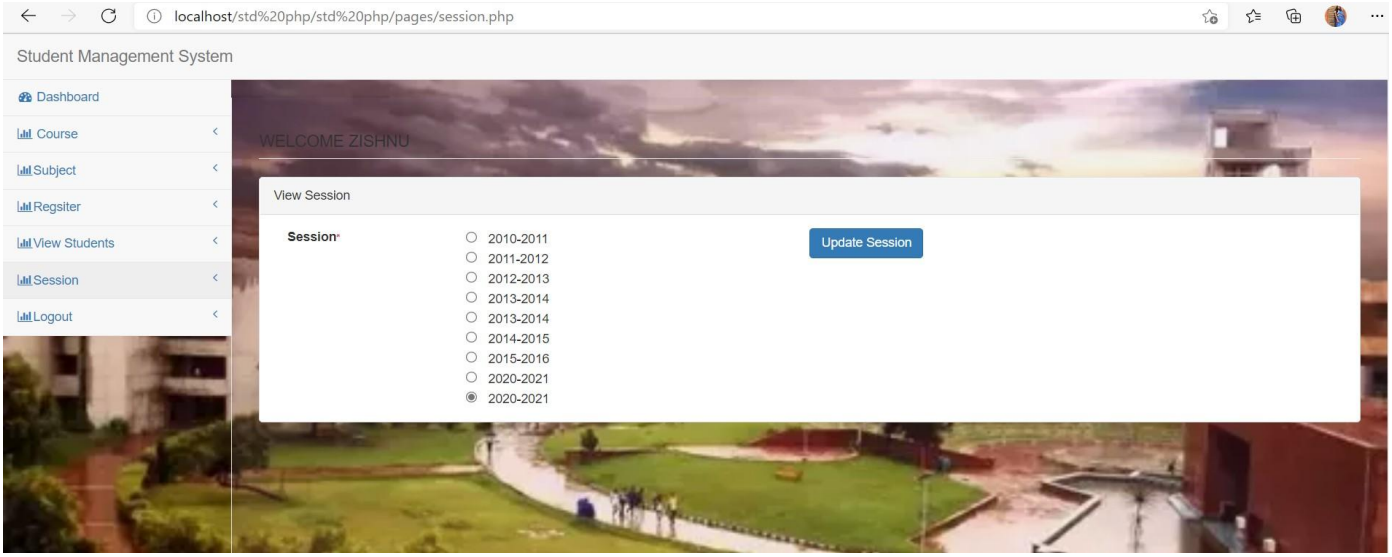


Fig : select session to see sessions activity

System and Database:

We have shown some parts of our codes where we have used SQL queries to form tables and import datas in the schemas. We also implemented those in myPHPadmin.

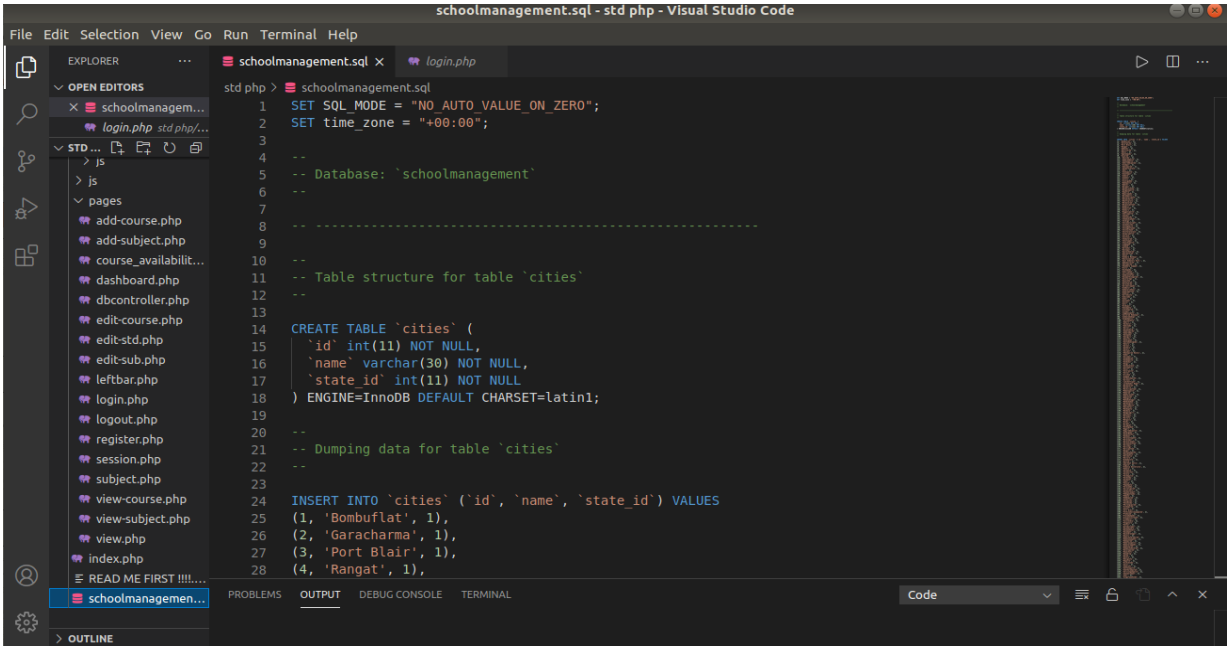
Here, we have used SQL Commands like,

- Create TABLE which is used to create the schemas. We have created 8tables such as Cities, Countries, Registration, Session, State, Subject, Course and Login table.
- Insert INTO is used to insert data into the table.
- Alter command is used to modify the tables which are already created such as adding a primary key, or adding a new tuple etc.
- We use here three datasets for cities , countries and states which are directly added as an option in the student registration form .

After identifying the tables and columns of the database, the next step is to create them. Basically there are two ways to do so. The first option is to create them using commands. Below are some of the ’create’ and ’insert’ statements. The complete database code is provided under Appendix B.

```
create table if not exists admin(aid char(50) not null primary key, aname char(50), apwd char(50));
 insert into admin values(’admin’, ’zishnu’, password(’sristi’));
```

The second option is to use phpMyAdmin. This is a Graphical User Interface (GUI) interface for building and maintaining the database which is included in the web server, XAMPP package. The first step is to create the database as shown in Figure below. The default storage engine in this MySQL server is InnoDB which has commit, rollback, and crash-recovery capabilities to protect user data. Following that is to create the tables in the database.



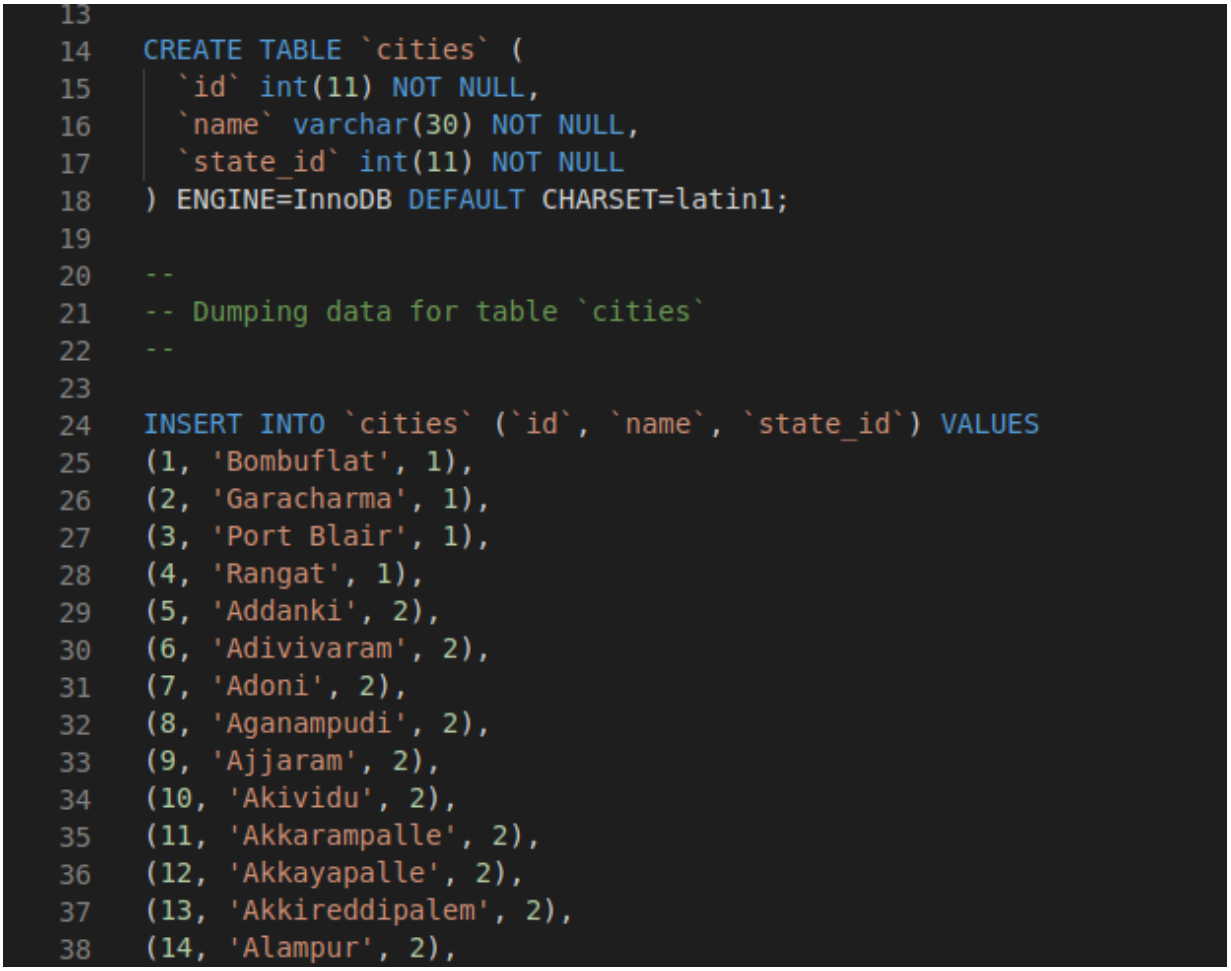


Fig: For the entity ‘Cities’ and inserting datas in the table

```
47620  (47573, 'Mvuma', 4120),
47621  (47574, 'Redcliffe', 4120),
47622  (47575, 'Shurugwi', 4120),
47623  (47576, 'Zvishavane', 4120);
47624
47625  -- --------------------------------------------------------
47626
47627  --
47628  -- Table structure for table `countries`
47629  --
47630
47631  CREATE TABLE `countries` (
47632    `id` int(11) NOT NULL,
47633    `sortname` varchar(50) NOT NULL,
47634    `name` varchar(100) NOT NULL
47635  ) ENGINE=InnoDB DEFAULT CHARSET=latin1;
47636
47637  --
47638  -- Dumping data for table `countries`
47639  --
47640
47641  INSERT INTO `countries` (`id`, `sortname`, `name`) VALUES
47642  (1, 'AF', 'Afghanistan'),
47643  (2, 'AL', 'Albania'),
47644  (3, 'DZ', 'Algeria'),
47645  (4, 'AS', 'American Samoa'),
47646  (5, 'AD', 'Andorra'),
```

```
47885  (244, 'YU', 'Yugoslavia'),
47886  (245, 'ZM', 'Zambia'),
47887  (246, 'ZW', 'Zimbabwe');
47888
47889  -- --------------------------------------------------------
47890
47891  --
47892  -- Table structure for table `registration`
47893  --
47894
47895  CREATE TABLE `registration` (
47896    `course` varchar(250) NOT NULL,
47897    `subject` varchar(250) NOT NULL,
47898    `fname` varchar(250) NOT NULL,
47899    `mname` varchar(250) NOT NULL,
47900    `lname` varchar(250) NOT NULL,
47901    `gender` varchar(50) NOT NULL,
47902    `gname` varchar(250) NOT NULL,
47903    `ocp` varchar(50) NOT NULL,
47904    `income` varchar(250) NOT NULL,
47905    `category` varchar(250) NOT NULL,
47906    `pchal` varchar(250) NOT NULL,
47907    `nationality` varchar(250) NOT NULL,
47908    `mobno` varchar(50) NOT NULL,
47909    `emailid` varchar(250) NOT NULL,
47910    `country` varchar(50) NOT NULL,
```

Fig : For the entity 'Countries' and inserting data in the table

```
47910     `country` varchar(50) NOT NULL,
47911     `state` varchar(50) NOT NULL,
47912     `dist` varchar(50) NOT NULL,
47913     `padd` text NOT NULL,
47914     `cadd` text NOT NULL,
47915     `board` varchar(50) NOT NULL,
47916     `roll` varchar(50) NOT NULL,
47917     `pyear` varchar(50) NOT NULL,
47918     `sub` varchar(250) NOT NULL,
47919     `marks` bigint(100) NOT NULL,
47920     `fmarks` bigint(100) NOT NULL,
47921     `board1` varchar(250) NOT NULL,
47922     `roll1` varchar(250) NOT NULL,
47923     `yop1` varchar(250) NOT NULL,
47924     `sub1` varchar(250) NOT NULL,
47925     `session` varchar(250) NOT NULL,
47926     `regdate` timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP,
47927     `marks1` varchar(50) NOT NULL,
47928     `fmarks1` varchar(50) NOT NULL,
47929     `regno` varchar(250) NOT NULL,
47930     `id` int(11) NOT NULL
47931   ) ENGINE=InnoDB DEFAULT CHARSET=latin1;
47932
47933   --
47934   -- Dumping data for table `registration`
47935   --
```

```
47935  --
47936
47937  INSERT INTO `registration` (`course`, `subject`, `fname`, `mname`, `lname`, `gender`, `gname`, `ocp`, `
47938  ('10', 'ENGINEERING DRAWING+COMPUTER PROGRAMMING+ COMPUTER SCIENCE ESSENTIALS', 'rahul', '', 'roy', 'fe
47939
47940  -- --------------------------------------------------------
47941
47942  --
47943  -- Table structure for table `session`
47944  --
47945
47946  CREATE TABLE `session` (
47947    `id` int(11) NOT NULL,
47948    `session` varchar(50) NOT NULL,
47949    `postingdate` date NOT NULL,
47950    `status` int(11) NOT NULL
47951  ) ENGINE=InnoDB DEFAULT CHARSET=latin1;
47952
47953  --
47954  -- Dumping data for table `session`
47955  --
47956
```

Fig: For entity registration

```
47956
47957   INSERT INTO `session` (`id`, `session`, `postingdate`, `status`) VALUES
47958   (1, '2010-2011', '2010-04-14', 0),
47959   (2, '2011-2012', '2012-04-14', 0),
47960   (3, '2012-2013', '2012-04-13', 0),
47961   (4, '2013-2014', '2013-04-05', 0),
47962   (5, '2013-2014', '2014-04-12', 0),
47963   (6, '2014-2015', '2015-04-12', 0),
47964   (7, '2015-2016', '2016-04-12', 0),
47965   (8, '2020-2021', '2021-04-12', 1);
47966
47967   -- --------------------------------------------------------
47968
47969   --
47970   -- Table structure for table `states`
47971   --
47972
47972
47973   CREATE TABLE `states` (
47974     `id` int(11) NOT NULL,
47975     `name` varchar(30) NOT NULL,
47976     `country_id` int(11) NOT NULL DEFAULT '1'
47977   ) ENGINE=InnoDB DEFAULT CHARSET=latin1;
47978
47979   --
47980   -- Dumping data for table `states`
47981   --
47982
47983   INSERT INTO `states` (`id`, `name`, `country_id`) VALUES
47984   (1, 'Andaman and Nicobar Islands', 101),
47985   (2, 'Andhra Pradesh', 101),
47986   (3, 'Arunachal Pradesh', 101),
47987   (4, 'Assam', 101),
47988   (5, 'Bihar', 101),
47989   (6, 'Chandigarh', 101),
47990   (7, 'Chhattisgarh', 101),
```

Fig: For the entity ‘states’ and inserting values

```
52103  (4119, 'Matabeleland South', 246),
52104  (4120, 'Midlands', 246);
52105
52106  -- --------------------------------------------------------
52107
52108  --
52109  -- Table structure for table `subject`
52110  --
52111
52112  CREATE TABLE `subject` (
52113    `subid` int(11) NOT NULL,
52114    `cshort` varchar(50) NOT NULL,
52115    `cfull` varchar(250) NOT NULL,
52116    `sub1` varchar(250) NOT NULL,
52117    `sub2` varchar(250) NOT NULL,
52118    `sub3` varchar(250) NOT NULL,
52119    `dt_created` datetime NOT NULL DEFAULT CURRENT_TIMESTAMP
52120  ) ENGINE=InnoDB DEFAULT CHARSET=latin1;
52121
52122  --
52123  -- Dumping data for table `subject`
52124  --
52125
52126  INSERT INTO `subject` (`subid`, `cshort`, `cfull`, `sub1`, `sub2`, `sub3`, `dt_created`) VALUES
52127  (3, '8', 'MASTER OF SCIENCE', 'Mathematics', 'Physics', 'Chemistry', '2016-04-16 18:08:27'),
52128  (6, '10', 'BACHELOR OF TECHNOLOGY', 'Engineering Drawing', 'Computer Programming', 'Computer Science Es
52129
52130  -- --------------------------------------------------------
```

```
52131
52132  --
52133  -- Table structure for table `tbl_course`
52134  --
52135
52136  CREATE TABLE `tbl_course` (
52137    `cid` int(11) NOT NULL,
52138    `cshort` varchar(250) NOT NULL,
52139    `cfull` varchar(250) NOT NULL,
52140    `cdate` varchar(50) NOT NULL
52141  ) ENGINE=InnoDB DEFAULT CHARSET=latin1;
52142
52143  --
52144  -- Dumping data for table `tbl_course`
52145  --
52146
52147  INSERT INTO `tbl_course` (`cid`, `cshort`, `cfull`, `cdate`) VALUES
52148  (7, 'M.C.A', 'MASTER OF COMPUTER APPLICATION', '11-04-2021'),
52149  (8, 'M.S.C', 'MASTER OF SCIENCE', '11-04-2021'),
52150  (9, 'B.Com', 'BACHELOR OF COMMERCE', '12-04-2021'),
52151  (10, 'B.TECH', 'BACHELOR OF TECHNOLOGY', '12-04-2021'),
52152  (11, 'M.B.A', 'MASTER OF BUSINESS ADMINISTRATION', '17-04-2021');
52153
```

Fig : For entity ‘Course’ and inserting values

```
52154   -- --------------------------------------------------------
52155
52156   --
52157   -- Table structure for table `tbl_login`
52158   --
52159
52160   CREATE TABLE `tbl_login` (
52161     `id` int(11) NOT NULL,
52162     `loginid` varchar(250) NOT NULL,
52163     `password` text NOT NULL
52164   ) ENGINE=InnoDB DEFAULT CHARSET=latin1;
52165
52166   --
52167   -- Dumping data for table `tbl_login`
52168   --
52169
52170   INSERT INTO `tbl_login` (`id`, `loginid`, `password`) VALUES
52171   (1, 'zishnu', 'sristi');
52172
52173   --
52174   -- Indexes for dumped tables
52175   --
52176
52177   --
52178   -- Indexes for table `cities`
52179   --
52180   ALTER TABLE `cities`
52181     ADD PRIMARY KEY (`id`);
52182
52183   --
52184   -- Indexes for table `countries`
52185   --
52186   ALTER TABLE `countries`
52187     ADD PRIMARY KEY (`id`);
52188
52189   --
52190   -- Indexes for table `registration`
52191   --
52192   ALTER TABLE `registration`
52193     ADD PRIMARY KEY (`id`);
52194
52195   --
52196   -- Indexes for table `states`
52197   --
52198   ALTER TABLE `states`
52199     ADD PRIMARY KEY (`id`);
52200
52201   --
52202   -- Indexes for table `subject`
52203   --
```

Fig : Creating the login entity

```
52209  --
52210  ALTER TABLE `tbl_course`
52211    ADD PRIMARY KEY (`cid`);
52212
52213  --
52214  -- Indexes for table `tbl_login`
52215  --
52216  ALTER TABLE `tbl_login`
52217    ADD PRIMARY KEY (`id`);
52218
52219  --
52220  -- AUTO_INCREMENT for dumped tables
52221  --
52222
52223  --
52224  -- AUTO_INCREMENT for table `cities`
52225  --
52226  ALTER TABLE `cities`
52227    MODIFY `id` int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=47577;
52228  --
52229  -- AUTO_INCREMENT for table `countries`
52230  --
52231  ALTER TABLE `countries`
52232    MODIFY `id` int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=247;
52233  --
52234  -- AUTO_INCREMENT for table `registration`
52235  --
```

```
52236  ALTER TABLE `registration`
52237    MODIFY `id` int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=3;
52238  --
52239  -- AUTO_INCREMENT for table `states`
52240  --
52241  ALTER TABLE `states`
52242    MODIFY `id` int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=4121;
52243  --
52244  -- AUTO_INCREMENT for table `subject`
52245  --
52246  ALTER TABLE `subject`
52247    MODIFY `subid` int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=7;
52248  --
52249  -- AUTO_INCREMENT for table `tbl_course`
52250  --
52251  ALTER TABLE `tbl_course`
52252    MODIFY `cid` int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=12;
52253  --
52254  -- AUTO_INCREMENT for table `tbl_login`
52255  --
52256  ALTER TABLE `tbl_login`
52257    MODIFY `id` int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=2;
52258
```

Fig: Modifying already existing tables by using Alter Command

To view the data in one of the tables, this can be done by clicking on the table name listed on the left menu as per Figure below. Whenever we write code in SQL the changes happening in the backend will also affect the phpMyAdmin . when a student will register or any course , subject etc will register , it will automatically saved in the server. phpMyAdmin is a free software tool written in PHP, intended to handle the administration of MySQL over the Web. phpMyAdmin supports a wide range of operations on MySQL and MariaDB. Frequently used operations (managing databases, tables, columns, relations, indexes, users, permissions, etc) can be performed via the user interface, while you still have the ability to directly execute any SQL statement . So, data will be saved here , we can edit any value of our database system through phpMyAdmin.

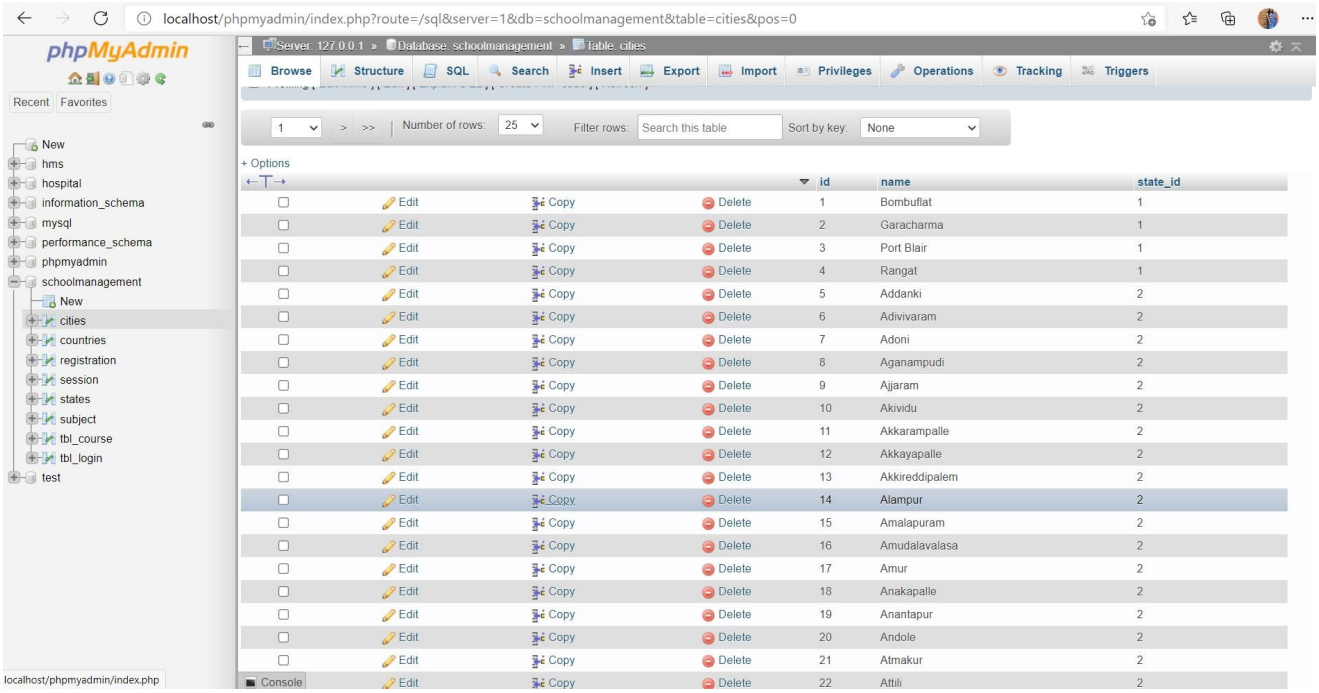


Figure: It is showing the database system along with the datas in the table Cities

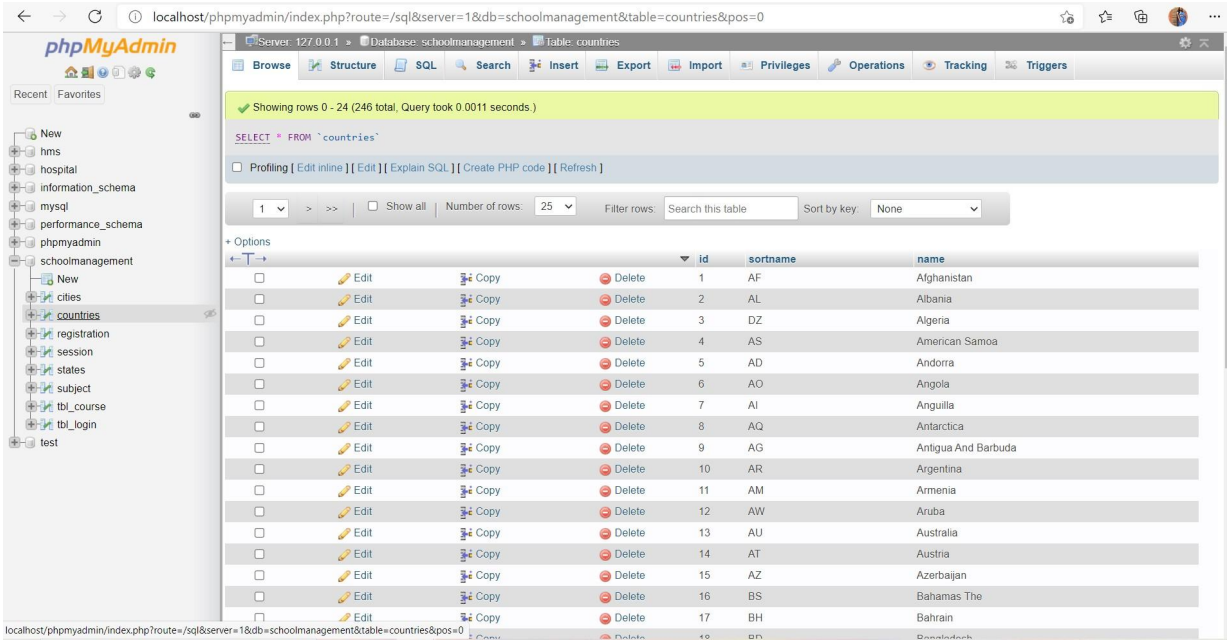


Figure: It is showing the database system along with the datas in the table countries

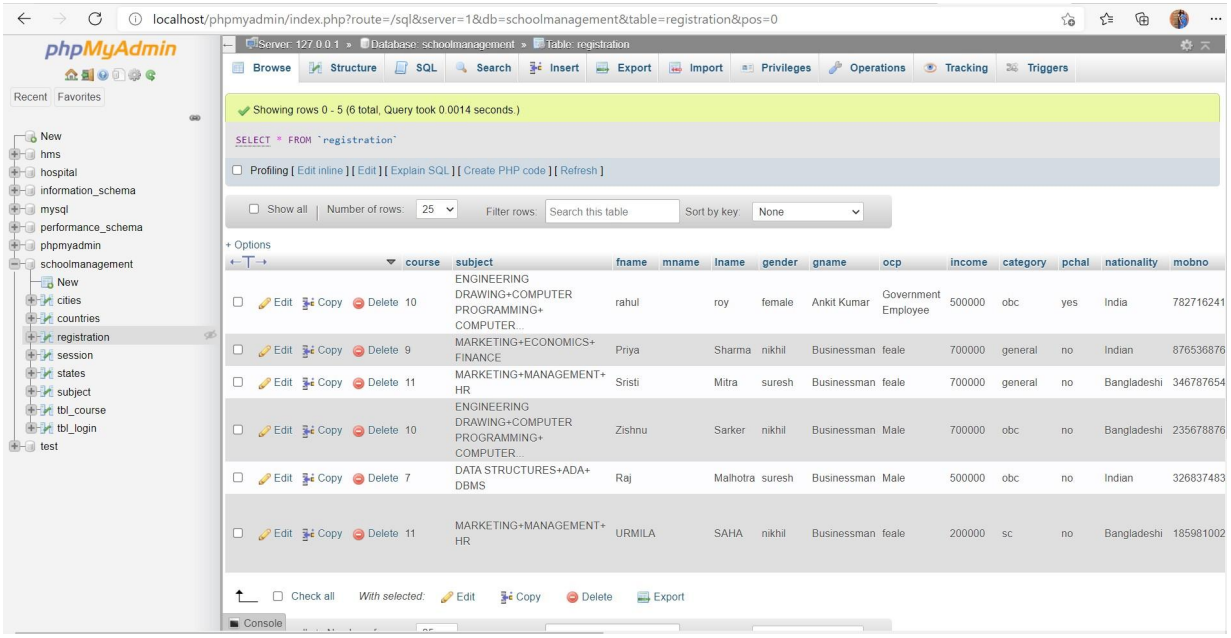


Figure: It is showing the database system along with the datas in the table Registration

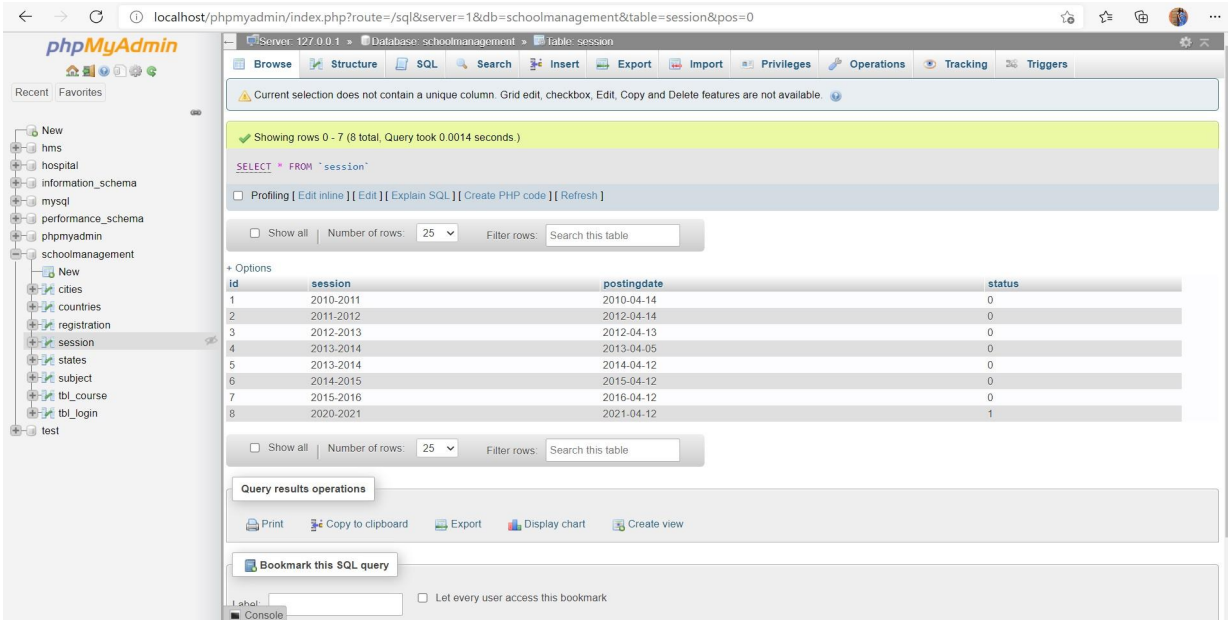


Figure: It is showing the database system along with the datas in the table Sessions

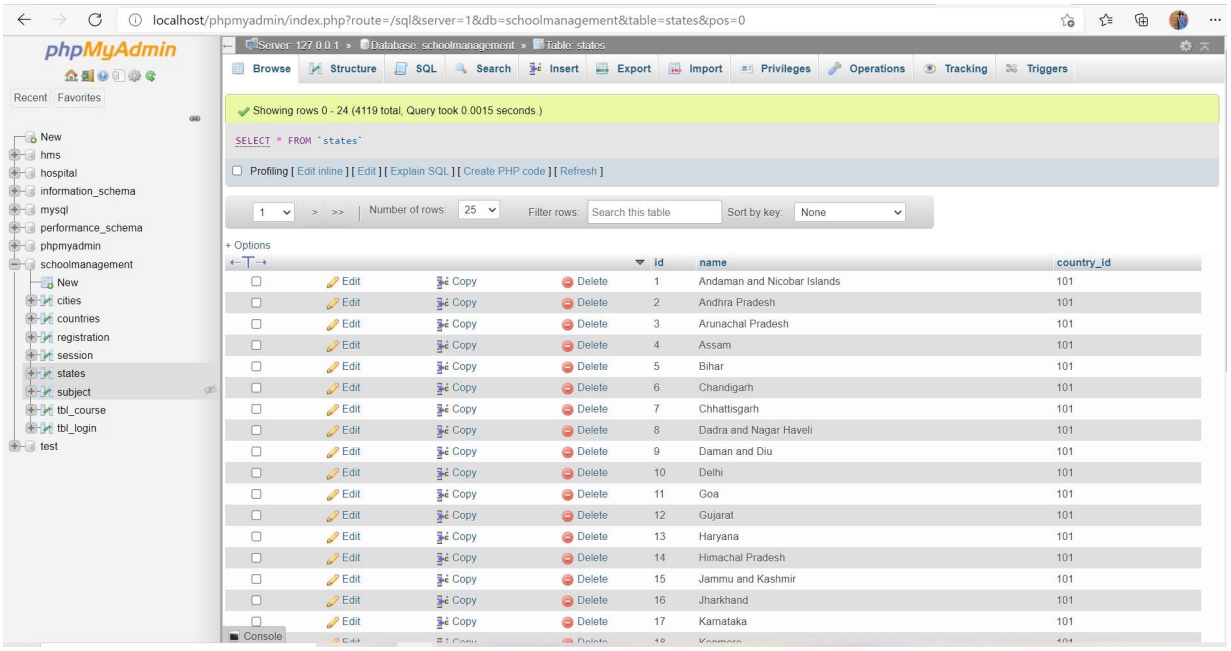


Figure: It is showing the database system along with the datas in the table States

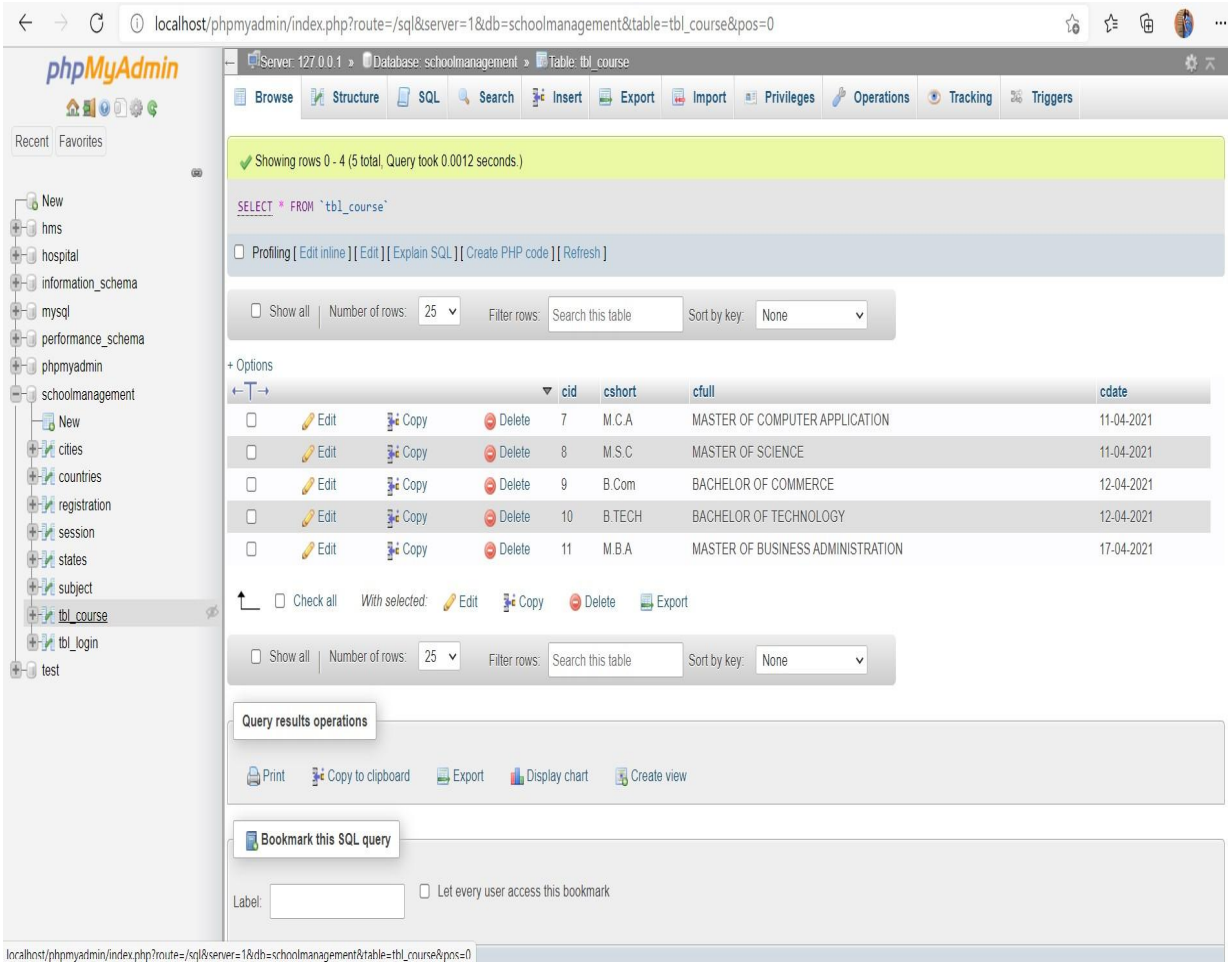
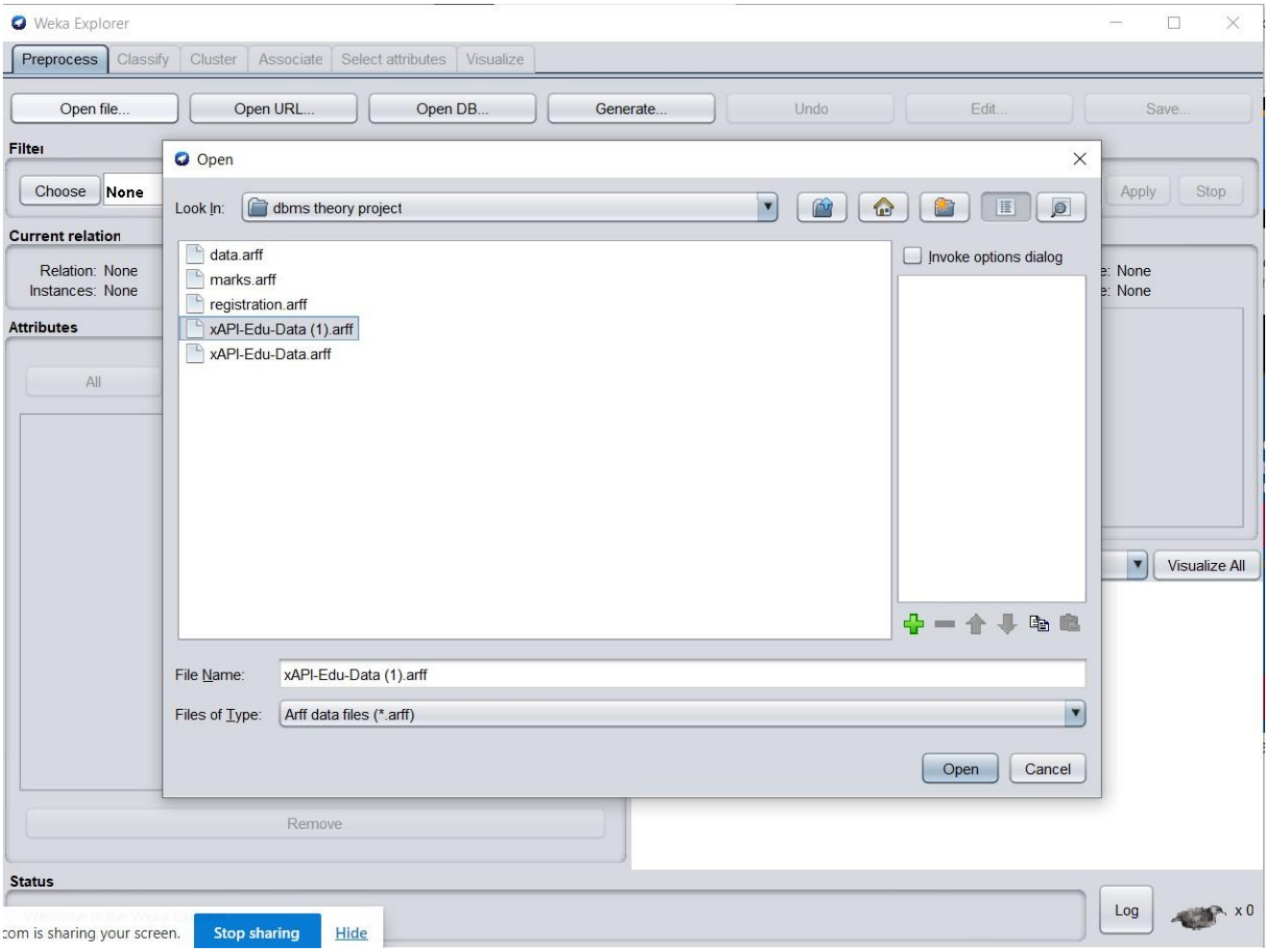


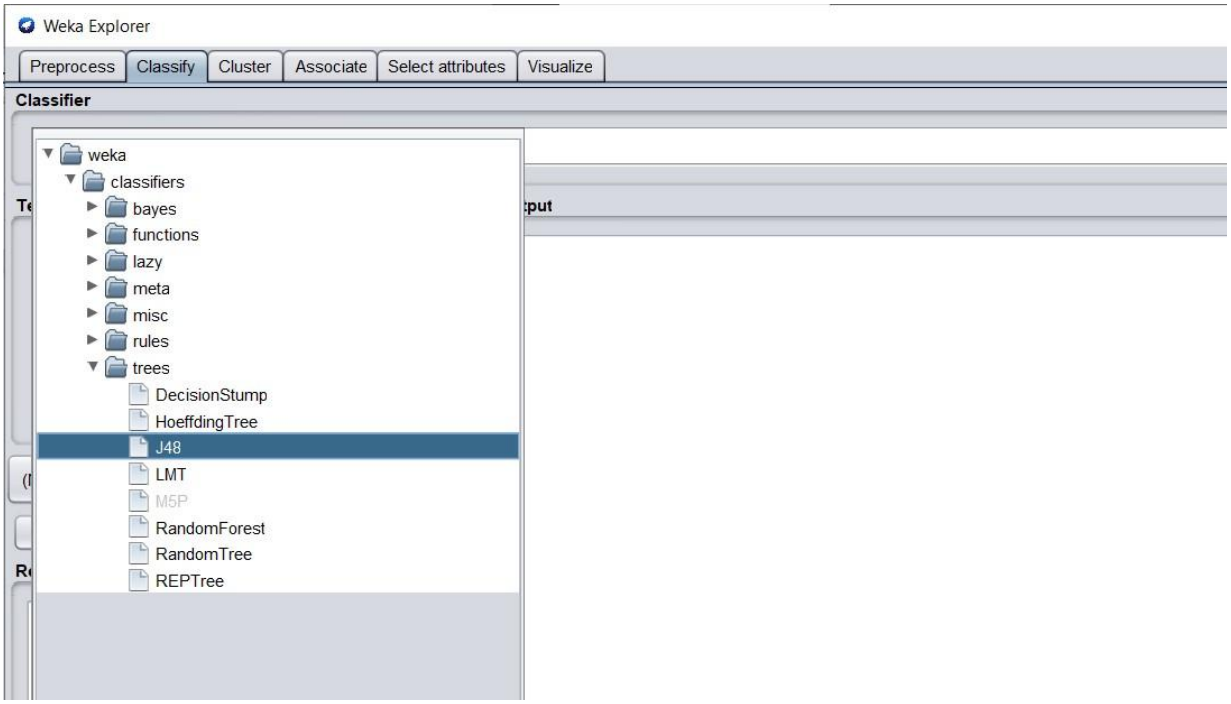
Figure: It is showing the database system along with the datas in the table course

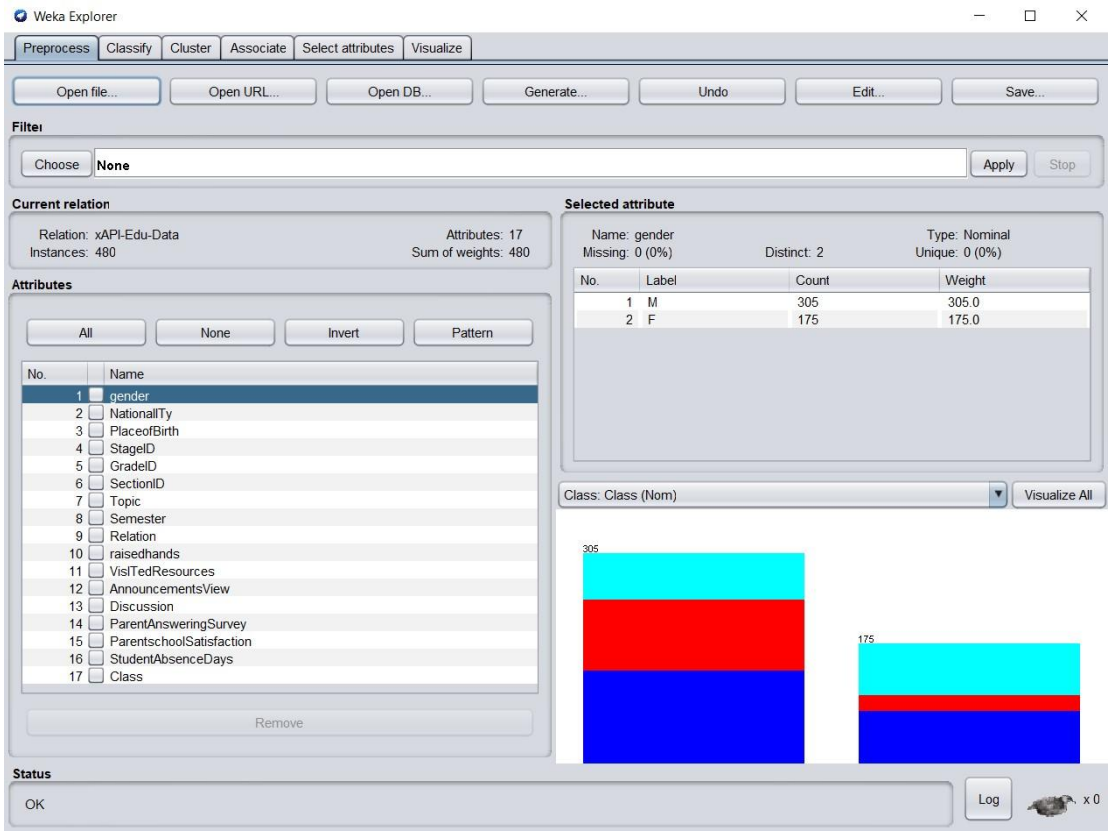
Result Prediction in WEKA :

At first we need to open Weka software > then we will tap explorer > after that we choose preprocessor from menus > and there we open the student.arff file > after that we shift to the classify option >there we choose weka > classifier > trees > J48 . Then it will provide us the good or bad prediction with confusion matrix , accuracy table and some percentages based on classified instances and some error percentages as well as it provides a decision tree which all will be elaborately discussed in the analysis part .

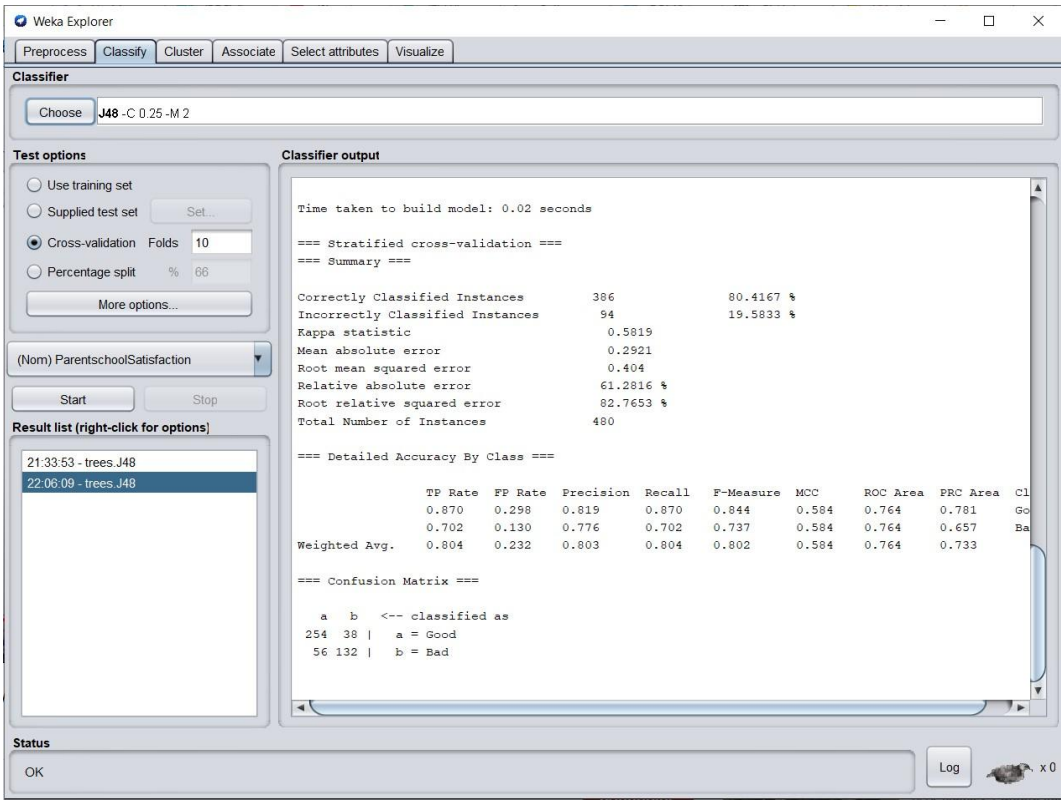

Weka GUI Chooser
Program Visualization Tools Help
Applications
Explorer
Experimenter
KnowledgeFlow
Workbench
Simple CLI
WEKA
The University of Waikato
Waikato Environment for Knowledge Analysis
Version 3.9.5
(c) 1999 - 2020
The University of Waikato
Hamilton, New Zealand
Weka, a native bird of New Zealand


Weka Explorer
Preprocess Classify Cluster Associate Select attributes Visualize
Open file... Open URL... Open DB... Generate... Undo Edit... Save...
Filter
Choose None
Apply Stop
Current relation
Relation: None
Instances: None
Attributes
All
Remove
Visualize All
Status
Log
x 0
Open
Look In: dbms theory project
data.arff
marks.arff
registration.arff
xAPI-Edu-Data (1).arff
xAPI-Edu-Data.arff
Invoke options dialog
File Name: xAPI-Edu-Data (1).arff
Files of Type: Arff data files (*.arff)
Open
Cancel
com is sharing your screen.
Stop sharing
Hide

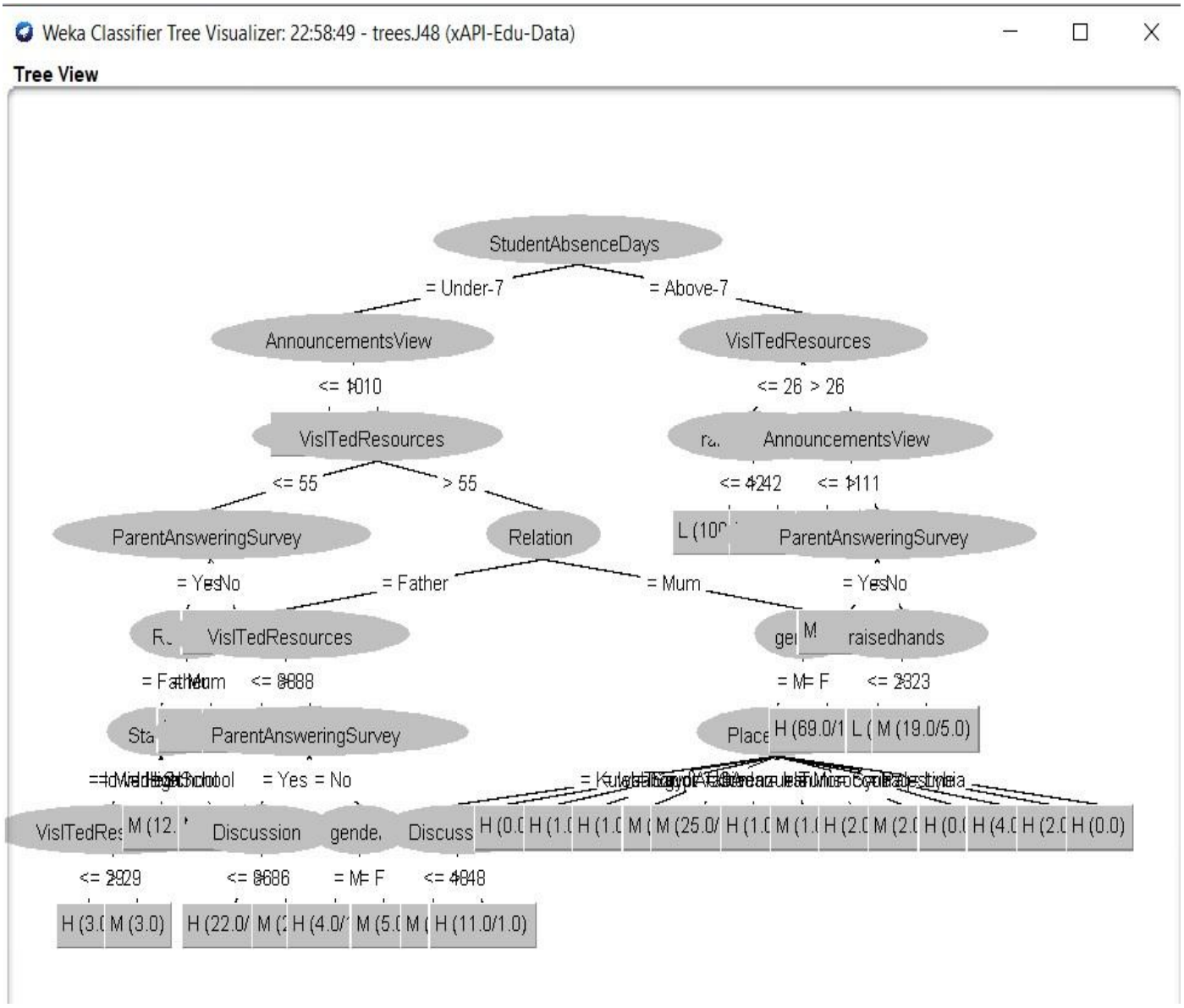

Weka Explorer
Preprocess
Classify
Cluster
Associate
Select attributes
Visualize
Classifier
weka
classifiers
bayes
functions
lazy
meta
misc
rules
trees
DecisionStump
HoeffdingTree
J48
LMT
M5P
RandomForest
RandomTree
REPTree


Weka Explorer
Preprocess
Classify
Cluster
Associate
Select attributes
Visualize
Open file...
Open URL...
Open DB...
Generate...
Undo
Edit...
Save...
Filter
Choose
None
Apply
Stop
Current relation
Relation: xAPI-Edu-Data
Instances: 480
Attributes: 17
Sum of weights: 480
Selected attribute
Name: gender
Missing: 0 (0%)
Distinct: 2
Type: Nominal
Unique: 0 (0%)
No. Label Count Weight
1 M 305 305.0
2 F 175 175.0
Attributes
All
None
Invert
Pattern
No. Name
1 gender
2 NationalITy
3 PlaceofBirth
4 StageID
5 GradeID
6 SectionID
7 Topic
8 Semester
9 Relation
10 raisedhands
11 VisITedResources
12 AnnouncementsView
13 Discussion
14 ParentAnsweringSurvey
15 ParentschoolSatisfaction
16 StudentAbsenceDays
17 Class
Remove
Class: Class (Nom)
Visualize All
305
175
Status
OK
Log
x 0

Once the model is generated, it can be used to classify new instances. Here classification is used to classify new instances using the J48 algorithm which is implemented in WEKA by the classifier. The classification model generation process using studentdata.arff in WEKA. Applying the same model to the test dataset. The running information of model generation, the running information of the test dataset. The graphical versions of the decision tree appeared then.



The student database contains 7 attributes such as personal information, physical information, academic information etc. For this analysis the relevant attributes such as academic year, institutional records which can predict the final term results of the student.



Experiment results show the efficiency and effectiveness of the J48 algorithm in predicting the Job-Position based on academic results. The model also can improve the efficiency of the results retrieving and evidently promote retrieval precision.

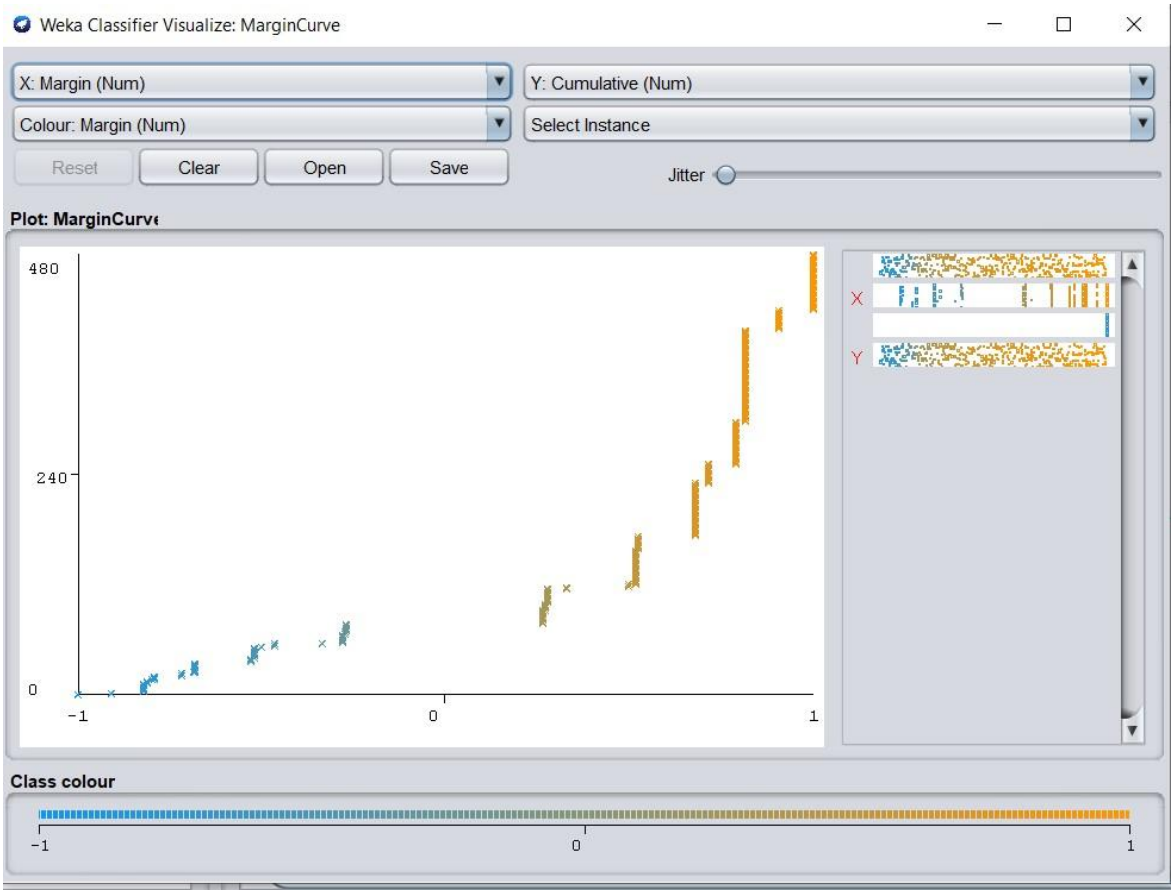


Figure: The marginal Curve of our prediction of the dataset

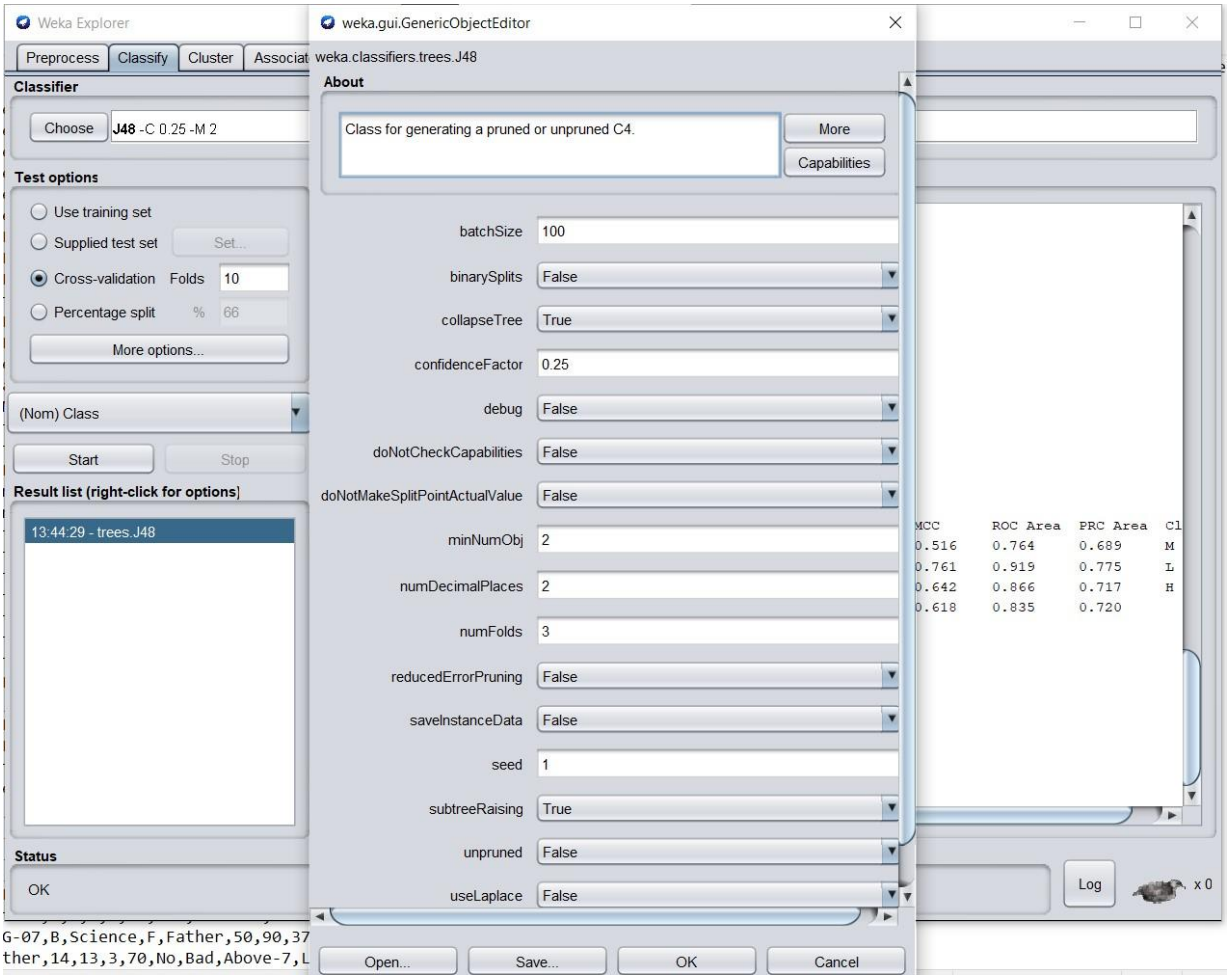
RESULT:

Discipline played an important role for the completion of this project. Using the project management skills taught in class, we were able to apply them to this project. Besides project management skills, we had also applied the database knowledge that we had learnt in the class to this project. From analysing of database structure and their relations to creating the database and tables, all these have strengthened my understanding. However when doing part of the codes, we would still research online and find solutions, for example, the usage of 'Ignore' command to avoid duplication of data from inserting into the database. We also had difficulty applying the J48 algorithm directly on our database, then we had to install Weka where we imported our database and it gave us the accurate results for the algorithm and prediction we needed.

Overall though the project was filled with challenges, it had given me opportunity to apply and strengthen my knowledge as well as gain new knowledge and skill through research and experienced professionals whom we are fortunate enough to encounter.

ANALYSIS:

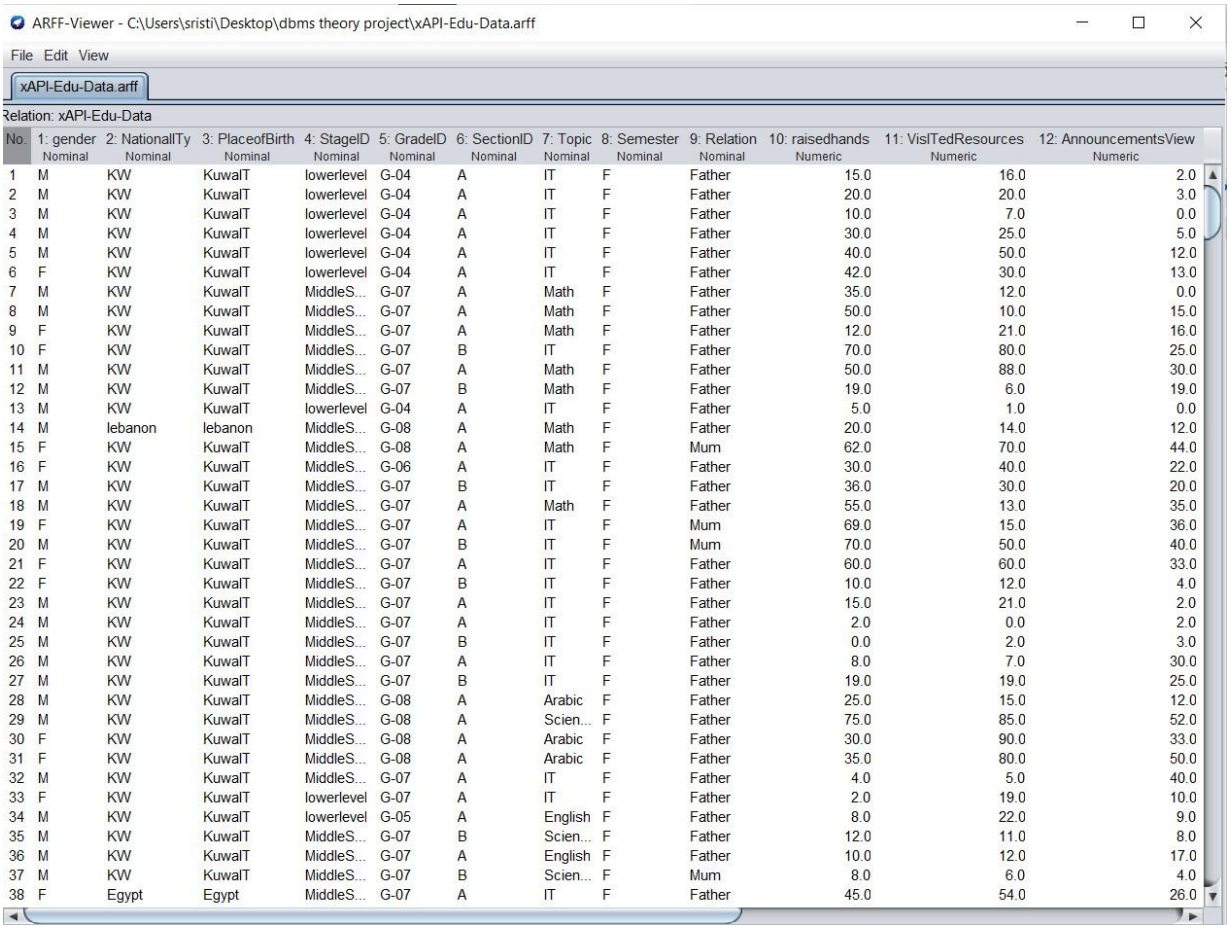
After the implementation, by analysing we get that our front end of the project runs properly as well as back end . To check our student database management project we downloaded the dataset from kaggle.com . Now in this part, we will analyze the prediction result of that database as we cant be able to add it directly to our database . So , we took the .arff file of the dataset and then ran it to the weka explorer . At the last part of analysis we will tell about our future work to use directly J48 inside the database and how we will develop our database with its beneficial site .



There we are working with the J48 algorithm which is a tree type . Like in the above fig , we can get into a new window by pressing the “ j48 -C 0.25 + M 2 “ which is a generic object

editor where we can edit and modify the code and parameters etc. But here we work with our dataset and its actual value , so we don't edit and modify anything .

There we have different ways of testing. We can use a training set and a separate test set based on our requirements. Training and testing is like dividing the data into a form where training an algorithm on a particular set of the data and we are testing on the rest of the part. So , if we have a separate test file we can choose it in the separate test set . In the cross validation area , we can choose the number of folds according to our requirements . and there is another option which is percentage split where we can divide the percentages of data of test set and training set . Also we can modify the classifier and choose which attribute we would like to apply in our classifier.



ARFF-Viewer - C:\Users\sristi\Desktop\dbms theory project\xAPI-Edu-Data.arff

File Edit View

xAPI-Edu-Data.arff

Relation: xAPI-Edu-Data

No.	1: gender Nominal	2: NationalITy Nominal	3: PlaceofBirth Nominal	4: StageID Nominal	5: GradeID Nominal	6: SectionID Nominal	7: Topic Nominal	8: Semester Nominal	9: Relation Nominal	10: raisedhands Numeric	11: VisITedResources Numeric	12: AnnouncementsView Numeric
1	M	KW	KuwaIT	lowerlevel	G-04	A	IT	F	Father	15.0	16.0	2.0
2	M	KW	KuwaIT	lowerlevel	G-04	A	IT	F	Father	20.0	20.0	3.0
3	M	KW	KuwaIT	lowerlevel	G-04	A	IT	F	Father	10.0	7.0	0.0
4	M	KW	KuwaIT	lowerlevel	G-04	A	IT	F	Father	30.0	25.0	5.0
5	M	KW	KuwaIT	lowerlevel	G-04	A	IT	F	Father	40.0	50.0	12.0
6	F	KW	KuwaIT	lowerlevel	G-04	A	IT	F	Father	42.0	30.0	13.0
7	M	KW	KuwaIT	MiddleS...	G-07	A	Math	F	Father	35.0	12.0	0.0
8	M	KW	KuwaIT	MiddleS...	G-07	A	Math	F	Father	50.0	10.0	15.0
9	F	KW	KuwaIT	MiddleS...	G-07	A	Math	F	Father	12.0	21.0	16.0
10	F	KW	KuwaIT	MiddleS...	G-07	B	IT	F	Father	70.0	80.0	25.0
11	M	KW	KuwaIT	MiddleS...	G-07	A	Math	F	Father	50.0	88.0	30.0
12	M	KW	KuwaIT	MiddleS...	G-07	B	Math	F	Father	19.0	6.0	19.0
13	M	KW	KuwaIT	lowerlevel	G-04	A	IT	F	Father	5.0	1.0	0.0
14	M	lebanon	lebanon	MiddleS...	G-08	A	Math	F	Father	20.0	14.0	12.0
15	F	KW	KuwaIT	MiddleS...	G-08	A	Math	F	Mum	62.0	70.0	44.0
16	F	KW	KuwaIT	MiddleS...	G-06	A	IT	F	Father	30.0	40.0	22.0
17	M	KW	KuwaIT	MiddleS...	G-07	B	IT	F	Father	36.0	30.0	20.0
18	M	KW	KuwaIT	MiddleS...	G-07	A	Math	F	Father	55.0	13.0	35.0
19	F	KW	KuwaIT	MiddleS...	G-07	A	IT	F	Mum	69.0	15.0	36.0
20	M	KW	KuwaIT	MiddleS...	G-07	B	IT	F	Mum	70.0	50.0	40.0
21	F	KW	KuwaIT	MiddleS...	G-07	A	IT	F	Father	60.0	60.0	33.0
22	F	KW	KuwaIT	MiddleS...	G-07	B	IT	F	Father	10.0	12.0	4.0
23	M	KW	KuwaIT	MiddleS...	G-07	A	IT	F	Father	15.0	21.0	2.0
24	M	KW	KuwaIT	MiddleS...	G-07	A	IT	F	Father	2.0	0.0	2.0
25	M	KW	KuwaIT	MiddleS...	G-07	B	IT	F	Father	0.0	2.0	3.0
26	M	KW	KuwaIT	MiddleS...	G-07	A	IT	F	Father	8.0	7.0	30.0
27	M	KW	KuwaIT	MiddleS...	G-07	B	IT	F	Father	19.0	19.0	25.0
28	M	KW	KuwaIT	MiddleS...	G-08	A	Arabic	F	Father	25.0	15.0	12.0
29	M	KW	KuwaIT	MiddleS...	G-08	A	Scien...	F	Father	75.0	85.0	52.0
30	F	KW	KuwaIT	MiddleS...	G-08	A	Arabic	F	Father	30.0	90.0	33.0
31	F	KW	KuwaIT	MiddleS...	G-08	A	Arabic	F	Father	35.0	80.0	50.0
32	M	KW	KuwaIT	MiddleS...	G-07	A	IT	F	Father	4.0	5.0	40.0
33	F	KW	KuwaIT	lowerlevel	G-07	A	IT	F	Father	2.0	19.0	10.0
34	M	KW	KuwaIT	lowerlevel	G-05	A	English	F	Father	8.0	22.0	9.0
35	M	KW	KuwaIT	MiddleS...	G-07	B	Scien...	F	Father	12.0	11.0	8.0
36	M	KW	KuwaIT	MiddleS...	G-07	A	English	F	Father	10.0	12.0	17.0
37	M	KW	KuwaIT	MiddleS...	G-07	B	Scien...	F	Mum	8.0	6.0	4.0
38	F	Egypt	Egypt	MiddleS...	G-07	A	IT	F	Father	45.0	54.0	26.0

fig: This is .arff file of our dataset

```
File  Edit  Format  View  Help
gender,NationalITy,PlaceofBirth,StageID,GradeID,SectionID,Topic,Semester,Relation,raisedhands,VisITedResources,AnnouncementsView,Discussion,ParentAnsweringSurvey,ParentschoolSatisfaction,St ^
udentAbsenceDays,Class
M,KW,KuwaIT,lowerlevel,G-04,A,IT,F,Father,15,16,2,20,Yes,Good,Under-7,M
M,KW,KuwaIT,lowerlevel,G-04,A,IT,F,Father,20,20,3,25,Yes,Good,Under-7,M
M,KW,KuwaIT,lowerlevel,G-04,A,IT,F,Father,10,7,0,30,No,Bad,Above-7,L
M,KW,KuwaIT,lowerlevel,G-04,A,IT,F,Father,30,25,5,35,No,Bad,Above-7,L
M,KW,KuwaIT,lowerlevel,G-04,A,IT,F,Father,40,50,12,50,No,Bad,Above-7,M
F,KW,KuwaIT,lowerlevel,G-04,A,IT,F,Father,42,30,13,70,Yes,Bad,Above-7,M
M,KW,KuwaIT,MiddleSchool,G-07,A,Math,F,Father,35,12,0,17,No,Bad,Above-7,L
M,KW,KuwaIT,MiddleSchool,G-07,A,Math,F,Father,50,10,15,22,Yes,Good,Under-7,M
F,KW,KuwaIT,MiddleSchool,G-07,A,Math,F,Father,12,21,16,50,Yes,Good,Under-7,M
F,KW,KuwaIT,MiddleSchool,G-07,B,IT,F,Father,70,80,25,70,Yes,Good,Under-7,M
M,KW,KuwaIT,MiddleSchool,G-07,A,Math,F,Father,50,88,30,80,Yes,Good,Under-7,H
M,KW,KuwaIT,MiddleSchool,G-07,B,Math,F,Father,19,6,19,12,Yes,Good,Under-7,M
M,KW,KuwaIT,lowerlevel,G-04,A,IT,F,Father,5,1,0,11,No,Bad,Above-7,L
M,lebanon,lebanon,MiddleSchool,G-08,A,Math,F,Father,20,14,12,19,No,Bad,Above-7,L
F,KW,KuwaIT,MiddleSchool,G-08,A,Math,F,Mum,62,70,44,60,No,Bad,Above-7,H
F,KW,KuwaIT,MiddleSchool,G-06,A,IT,F,Father,30,40,22,66,Yes,Good,Under-7,M
M,KW,KuwaIT,MiddleSchool,G-07,B,IT,F,Father,36,30,20,80,No,Bad,Above-7,M
M,KW,KuwaIT,MiddleSchool,G-07,A,Math,F,Father,55,13,35,90,No,Bad,Above-7,M
F,KW,KuwaIT,MiddleSchool,G-07,A,IT,F,Mum,69,15,36,96,Yes,Good,Under-7,M
M,KW,KuwaIT,MiddleSchool,G-07,B,IT,F,Mum,70,50,40,99,Yes,Good,Under-7,H
F,KW,KuwaIT,MiddleSchool,G-07,A,IT,F,Father,60,60,33,90,No,Bad,Above-7,M
F,KW,KuwaIT,MiddleSchool,G-07,B,IT,F,Father,10,12,4,80,No,Bad,Under-7,M
M,KW,KuwaIT,MiddleSchool,G-07,A,IT,F,Father,15,21,2,90,No,Bad,Under-7,M
M,KW,KuwaIT,MiddleSchool,G-07,A,IT,F,Father,2,0,2,50,No,Bad,Above-7,L
M,KW,KuwaIT,MiddleSchool,G-07,B,IT,F,Father,0,2,3,70,Yes,Good,Above-7,L
M,KW,KuwaIT,MiddleSchool,G-07,A,IT,F,Father,8,7,30,40,Yes,Good,Above-7,L
M,KW,KuwaIT,MiddleSchool,G-07,B,IT,F,Father,19,19,25,40,Yes,Bad,Under-7,M
M,KW,KuwaIT,MiddleSchool,G-08,A,Arabic,F,Father,25,15,12,33,No,Bad,Above-7,L
M,KW,KuwaIT,MiddleSchool,G-08,A,Science,F,Father,75,85,52,43,Yes,Good,Under-7,M
F,KW,KuwaIT,MiddleSchool,G-08,A,Arabic,F,Father,30,90,33,35,No,Bad,Under-7,M
F,KW,KuwaIT,MiddleSchool,G-08,A,Arabic,F,Father,35,80,50,70,Yes,Good,Under-7,H
M,KW,KuwaIT,MiddleSchool,G-07,A,IT,F,Father,4,5,40,16,Yes,Good,Above-7,L
F,KW,KuwaIT,lowerlevel,G-07,A,IT,F,Father,2,19,10,50,Yes,Good,Above-7,L
M,KW,KuwaIT,lowerlevel,G-05,A,English,F,Father,8,22,9,40,No,Bad,Above-7,L
M,KW,KuwaIT,MiddleSchool,G-07,B,Science,F,Father,12,11,8,40,No,Bad,Above-7,L
M,KW,KuwaIT,MiddleSchool,G-07,A,English,F,Father,10,12,17,30,No,Bad,Above-7,L
M,KW,KuwaIT,MiddleSchool,G-07,B,Science,F,Mum,8,6,4,22,Yes,Good,Above-7,L
```

fig ii : Training Set Instance (.arff File)

We use the .arff of our dataset as our training set , and after that we choose a separate test set where we took the same arff file but this time we did a percentage split where we provide 66% of our data is used for the training set and 34 % is for the test set . And after that , we start both the training set and the test set. And after that, the result comes in that the classifier output screen of weka software is not fully visible . So , **we copy paste it after fig iii (training dataset) and fig iv (test dataset)which are given below with its explanation inside comment (// …….. //) statement** .

The classification model generation process using studentdata.arff in WEKA is shown in Figure iii. Applying the same model to the test dataset is shown in Figure v. The running information of model generation is given in Figure iv, the running information of the test dataset is given in Figure vi.
The graphical versions of the decision tree appear in Figure vii and Figure viii.
Once the model is generated, the details of the model such as total number of instance in the dataset, the number of correctly classified instance, incorrectly classified instance,
and accuracy of the model will be displayed

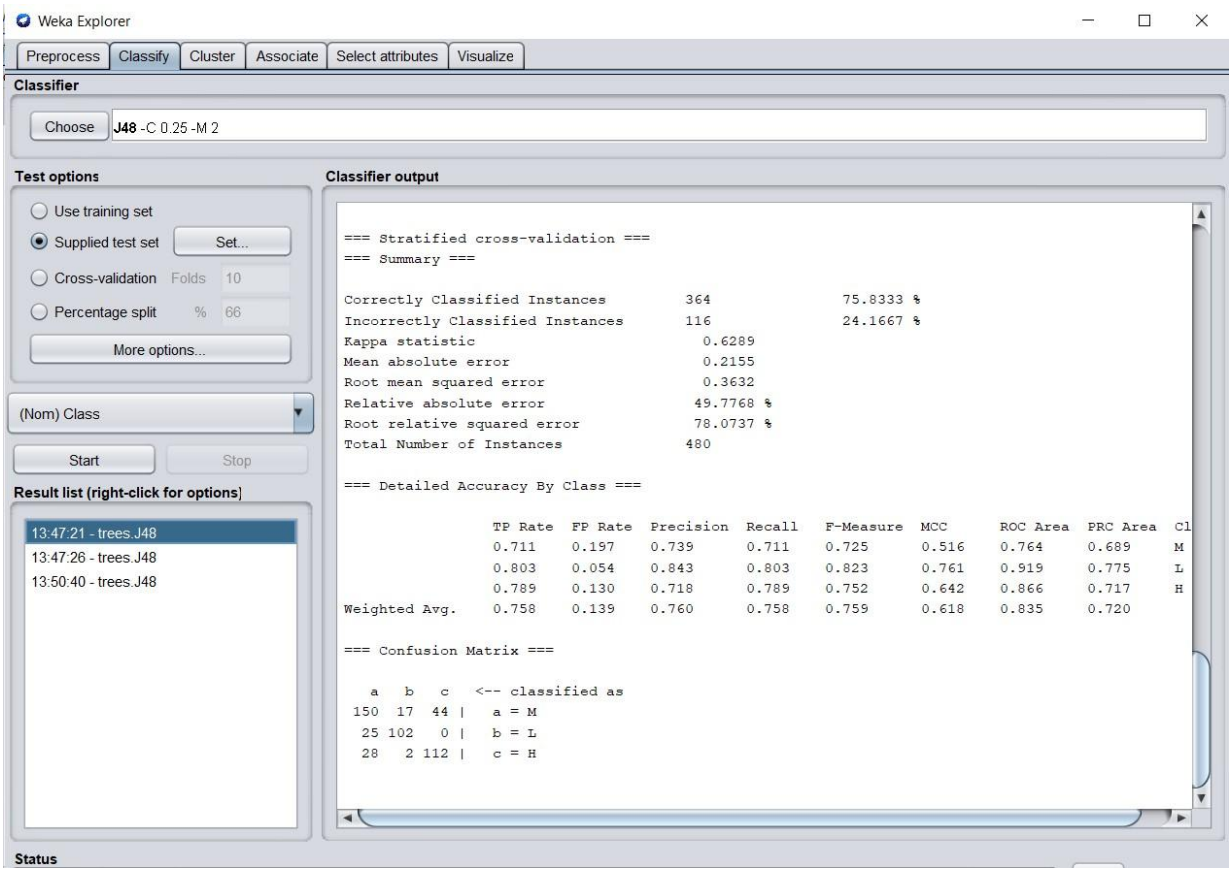


Fig iii: Applying C4.5 (J48) classifier, WEKA for Training Set – Screenshot

Classifier Model(Training Dataset)

=== Run information ===

Scheme: weka.classifiers.trees.J48 -C 0.25 -M 2

Relation: xAPI-Edu-Data

Instances: 480 // there are 480 instances in the training dataset //

Attributes: 17 // there are 17 attributes, as we can see them below //

gender

NationalITy

PlaceofBirth

StageID

GradeID

SectionID

Topic

Semester

Relation

raisedhands

VisITedResources

AnnouncementsView

Discussion

ParentAnsweringSurvey

ParentschoolSatisfaction

StudentAbsenceDays

Class

Test mode: 10-fold cross-validation // this is the test mode where we choose the number of folds of cross validation , the default value is 10 folds cross validation //

=== Classifier model (full training set) ===

J48 pruned tree //below there is pruned tree of the same
…….. training dataset //

StudentAbsenceDays = Under-7

AnnouncementsView <= 10: M (26.0/11.0)

AnnouncementsView > 10

VisITedResources <= 55

ParentAnsweringSurvey = Yes

Relation = Father

StageID = lowerlevel

VisITedResources <= 29: H (3.0)

VisITedResources > 29: M (3.0)

StageID = MiddleSchool: M (12.0)

StageID = HighSchool: M (4.0)

Relation = Mum: H (11.0/4.0)

ParentAnsweringSurvey = No: M (25.0/4.0)

VisITedResources > 55

Relation = Father

VisITedResources <= 88: M (50.0/12.0)

VisITedResources > 88

ParentAnsweringSurvey = Yes

Discussion <= 86: H (22.0/1.0)

Discussion > 86: M (2.0)

ParentAnsweringSurvey = No

gender = M: H (4.0/1.0)

gender = F: M (5.0)

Relation = Mum

gender = M

PlaceofBirth = KuwaIT

Discussion <= 48: M (2.0)

Discussion > 48: H (11.0/1.0)

PlaceofBirth = lebanon: H (0.0)

PlaceofBirth = Egypt: H (1.0)

PlaceofBirth = SaudiArabia: H (1.0)

PlaceofBirth = USA: M (1.0)

PlaceofBirth = Jordan: M (25.0/9.0)

PlaceofBirth = venzuela: H (1.0)

PlaceofBirth = Iran: M (1.0)

PlaceofBirth = Tunis: H (2.0)

PlaceofBirth = Morocco: M (2.0)

PlaceofBirth = Syria: H (0.0)

PlaceofBirth = Iraq: H (4.0)

PlaceofBirth = Palestine: H (2.0)

PlaceofBirth = Lybia: H (0.0)

gender = F: H (69.0/11.0)

StudentAbsenceDays = Above-7

VisITedResources <= 26

raisedhands <= 42: L (100.0/9.0)

raisedhands > 42: M (3.0/1.0)

VisITedResources > 26

AnnouncementsView <= 11: L (9.0)

AnnouncementsView > 11

ParentAnsweringSurvey = Yes: M (53.0/7.0)

ParentAnsweringSurvey = No

raisedhands <= 23: L (7.0)

raisedhands > 23: M (19.0/5.0)

Number of Leaves : 34 // here we can see the number of leaves of .
the tree which is 34//

Size of the tree : 54 // here we can see the size of the tree which .
which is 54 //

Time taken to build model: 0.01 seconds

=== Stratified cross-validation ===

=== Summary ===

Correctly Classified Instances 364 75.8333 % // there are 364 .
instances who are correctly classified which indicates 75.8333% accuracy //

Incorrectly Classified Instances 116 24.1667 % // there are 116 instances
who are incorrectly classified which indicates 24.1667% inaccuracy //

Kappa statistic 0.6289

Mean absolute error 0.2155

Root mean squared error 0.3632

Relative absolute error 49.7768 %

Root relative squared error 78.0737 %

Total Number of Instances 480

=== Detailed Accuracy By Class ===

	TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC-Area	PRC Area	Class
	0.711	0.197	0.739	0.711	0.725	0.516	0.764	0.689	M
	0.803	0.054	0.843	0.803	0.823	0.761	0.919	0.775	L
	0.789	0.130	0.718	0.789	0.752	0.642	0.866	0.717	H
Weighted Avg.	0.758	0.139	0.760	0.758	0.759	0.618	0.835	0.720	

// in the above detailed accuracy by class section , it is providing true-false rate , false-true rate , precision , recall , f-measure, MCC, Roc-area, PRC area , class . This is showing us the rate where it provides how many true are showing as false , how many false are showing true and according to that we can develop our algorithm and get more correct results. These are different parameters in the basis of which analyze the output of the classification algorithm //

=== Confusion Matrix ===

a	b	c	<-- classified as
150	17	44	a = M
25	102	0	b = L
28	2	112	c = H

Fig: 5 Training Dataset - Running InformationTest mode :10-fold cross-validation : size unknown (reading incrementally)

// there are 3 types of possibilities who are high , medium , low .

So the algorithm classified in the first case as in medium ranked result where 150 of them classified correctly as true as they will at least medium type of result and 17 of them will do lowest rank result and 44 of them will be in the list of highest ranked result .

In the second case as in lower ranked result , 25 is misclassified that they will do medium ranked result and 102 is also correctly classified as they cannot do lower ranked result . and at last misclassified as nobody will do the high ranked result .

In the third case , 28 and 2 are misclassified as they can do medium ranked result and lower ranked result respectively but 112 is correctly classified as they will not do high ranked result .

So if we see the diagonal of the confusion matrix, 150 +102 +112 is the total which we will get by looking at the correctly classified instances = 364 and the other side diagonal is 116 which is incorrectly classified instances. //

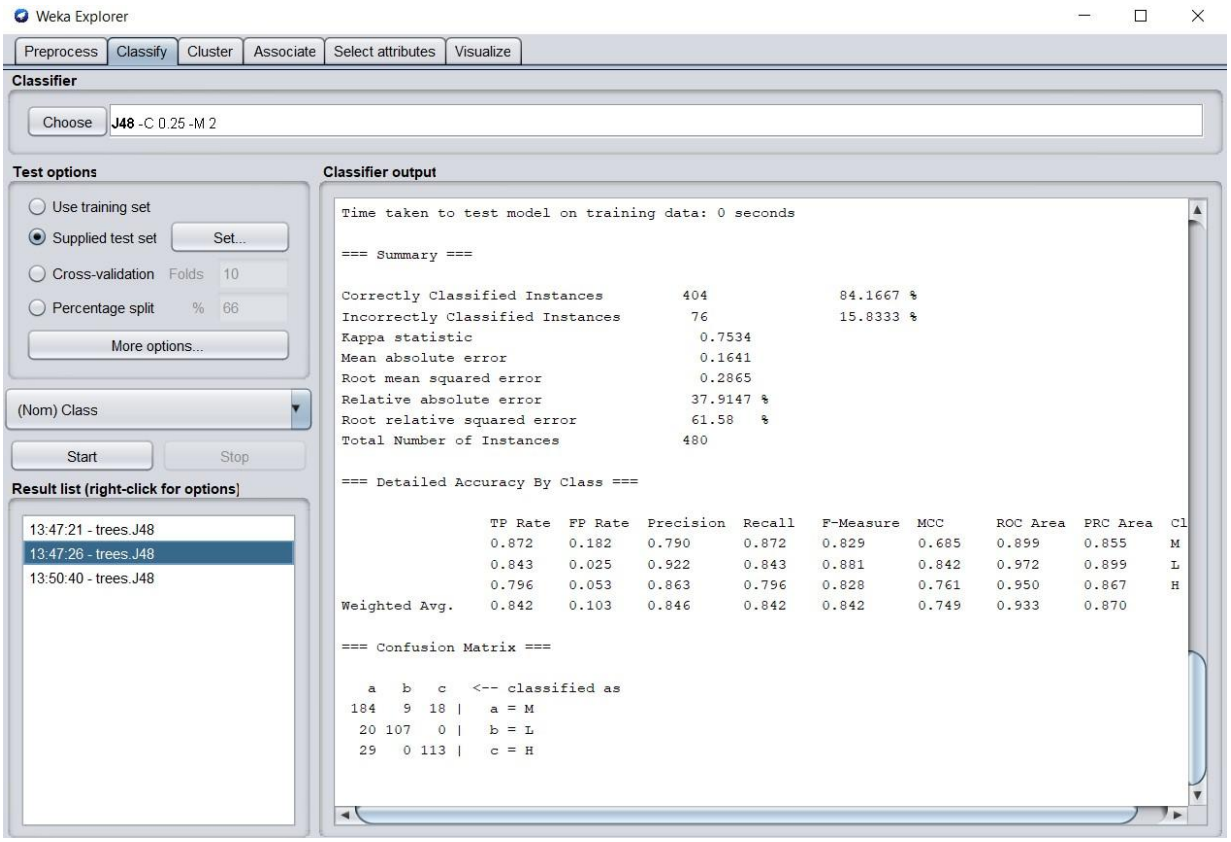


Fig v: Applying C4.5 (J48) classifier, WEKA – for Test set - Screenshot

=== Run information ===

Scheme: weka.classifiers.trees.J48 -C 0.25 -M 2

Relation: xAPI-Edu-Data

Instances: 480 // there are 480 instances in the training dataset //

Attributes: 17 // there are 17 attributes , as we can see below //

gender

NationalITy

PlaceofBirth

StageID

GradeID

SectionID

Topic

Semester

Relation

raisedhands

VisITedResources

AnnouncementsView

Discussion

ParentAnsweringSurvey

ParentschoolSatisfaction

StudentAbsenceDays

Class

Test mode: 10-fold cross-validation // this is the test mode where we choose the number of folds of cross validation , the default value is 10 folds cross validation //

=== Classifier model (full training set) ===

J48 pruned tree // below there is a pruned tree of the

---------- same test dataset//

StudentAbsenceDays = Under-7

AnnouncementsView <= 10: M (26.0/11.0)

AnnouncementsView > 10

VisITedResources <= 55

ParentAnsweringSurvey = Yes

Relation = Father

StageID = lowerlevel

VisITedResources <= 29: H (3.0)

VisITedResources > 29: M (3.0)

StageID = MiddleSchool: M (12.0)

StageID = HighSchool: M (4.0)

Relation = Mum: H (11.0/4.0)

ParentAnsweringSurvey = No: M (25.0/4.0)

VisITedResources > 55

Relation = Father

VisITedResources <= 88: M (50.0/12.0)

VisITedResources > 88

ParentAnsweringSurvey = Yes

Discussion <= 86: H (22.0/1.0)

Discussion > 86: M (2.0)

ParentAnsweringSurvey = No

gender = M: H (4.0/1.0)

gender = F: M (5.0)

Relation = Mum

gender = M

PlaceofBirth = KuwaIT

Discussion <= 48: M (2.0)

Discussion > 48: H (11.0/1.0)

PlaceofBirth = lebanon: H (0.0)

PlaceofBirth = Egypt: H (1.0)

PlaceofBirth = SaudiArabia: H (1.0)

PlaceofBirth = USA: M (1.0)

PlaceofBirth = Jordan: M (25.0/9.0)

PlaceofBirth = venzuela: H (1.0)

PlaceofBirth = Iran: M (1.0)

PlaceofBirth = Tunis: H (2.0)

PlaceofBirth = Morocco: M (2.0)

PlaceofBirth = Syria: H (0.0)

PlaceofBirth = Iraq: H (4.0)

PlaceofBirth = Palestine: H (2.0)

PlaceofBirth = Lybia: H (0.0)

gender = F: H (69.0/11.0)

StudentAbsenceDays = Above-7

VisITedResources <= 26

raisedhands <= 42: L (100.0/9.0)

raisedhands > 42: M (3.0/1.0)

VisITedResources > 26

AnnouncementsView <= 11: L (9.0)

AnnouncementsView > 11

ParentAnsweringSurvey = Yes: M (53.0/7.0)

ParentAnsweringSurvey = No

raisedhands <= 23: L (7.0)

raisedhands > 23: M (19.0/5.0)

Number of Leaves : 34 // here we can see the number of leaves of .
the tree which is 34//

Size of the tree : 54 // here we can see the size of .
the tree which is 54//

Time taken to test model on training data: 0 seconds

=== Summary ===

Correctly Classified Instances 404 84.1667 % // there are 404 . instances who are correctly classified which indicates 84.1667% accuracy //

Incorrectly Classified Instances 76 15.8333 % // there are 76 instances who are incorrectly classified which indicates 15.8333% inaccuracy //

Kappa statistic 0.7534

Mean absolute error 0.1641

Root mean squared error 0.2865

Relative absolute error 37.9147 %

Root relative squared error 61.58 %

Total Number of Instances 480

=== Detailed Accuracy By Class ===

	TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC-Area	PRC-Area	Class
	0.872	0.182	0.790	0.872	0.829	0.685	0.899	0.855	M
	0.843	0.025	0.922	0.843	0.881	0.842	0.972	0.899	L
	0.796	0.053	0.863	0.796	0.828	0.761	0.950	0.867	H
Weighted Avg.	0.842	0.103	0.846	0.842	0.842	0.749	0.933	0.870	

// in the above detailed accuracy by class section , it is providing true-false rate , false-true rate , precision , recall , f-measure, MCC, Roc-area, PRC area , class . This is showing us the rate where it provides how many true are showing as false , how many false are showing true and according to that we can develop our algorithm and get more correct results. These are different parameters in the basis of which analyze the output of the classification algorithm //

=== Confusion Matrix ===

```
  a   b   c   <-- classified as
184   9  18 |   a = M
 20 107   0 |   b = L
 29   0 113 |   c = H
```

Fig: 6 Running Information Test Dataset

In the above confusion matrix,it is classified as there are 3 types of possibilities who are high ranked status , medium ranked status , low ranked status .

So the algorithm classified in the first case as in medium ranked result where 184 of them classified correctly as true as they will at least medium type of result and 9 of them will do lowest rank result and 18 of them will be in the list of highest ranked result .

In the second case as in lower ranked result , 20 is misclassified that they will do medium ranked result and 107 is also correctly classified as they cannot do lower ranked result . and at last misclassified as nobody will do the high ranked result .

In the third case , 29 and 0 are misclassified as they can do medium ranked result and lower ranked result respectively but 113 is correctly classified as they will not do high ranked result .

So if we see the diagonal of the confusion matrix, 184 +107 +113 is the total which we will get by looking at the correctly classified instances = 404 and the other side diagonal is 76 which is incorrectly classified instances.

So here we get a more modified result where 404 instances are correctly classified where in the training set it has 364 correctly classified instances . This happens because the training set contains a lot of attributes and a lot of values but in the test set we reduce the number of attributes and values and run it . So , if we modify our database arff file and just take name , department , result , and academic year , it will predict more accurately how your future result will be according to the previous result . Based on the prediction result the institute, student, admin can aware of their future situation and take necessary steps according to the situation

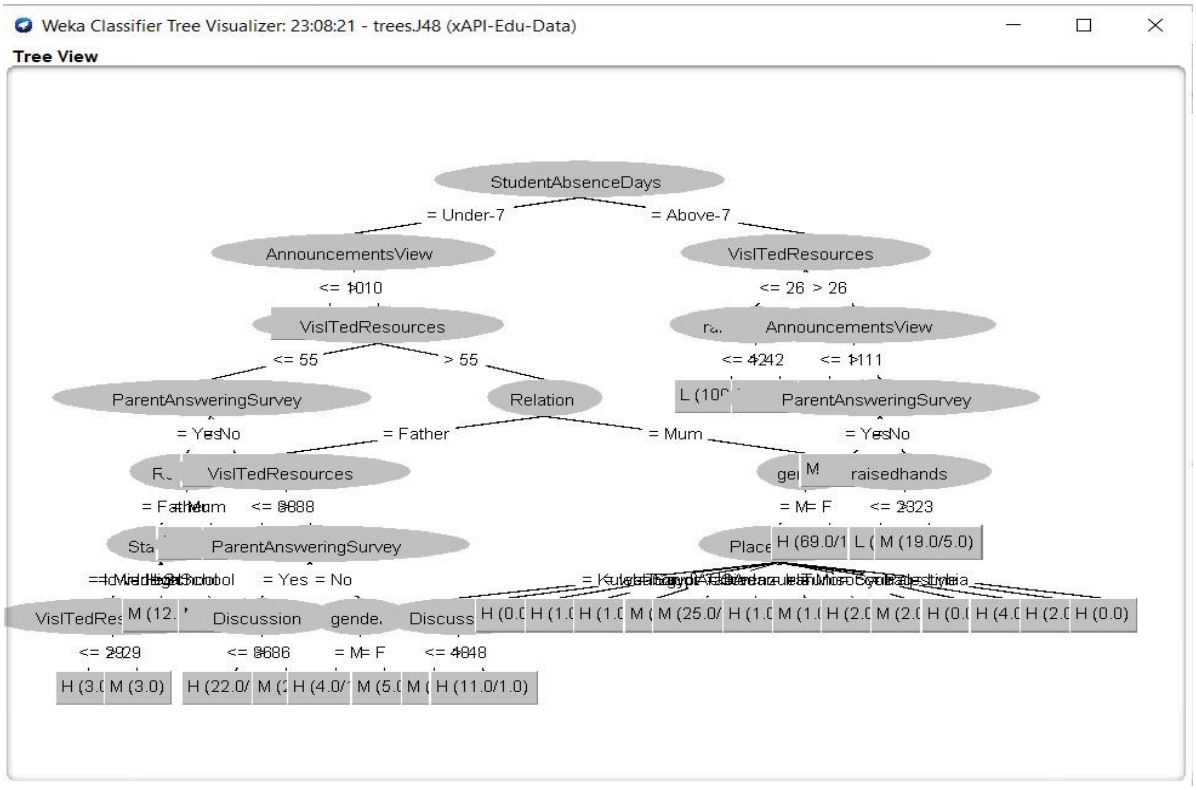


Fig : 7 - Decision Tree - for training dataset

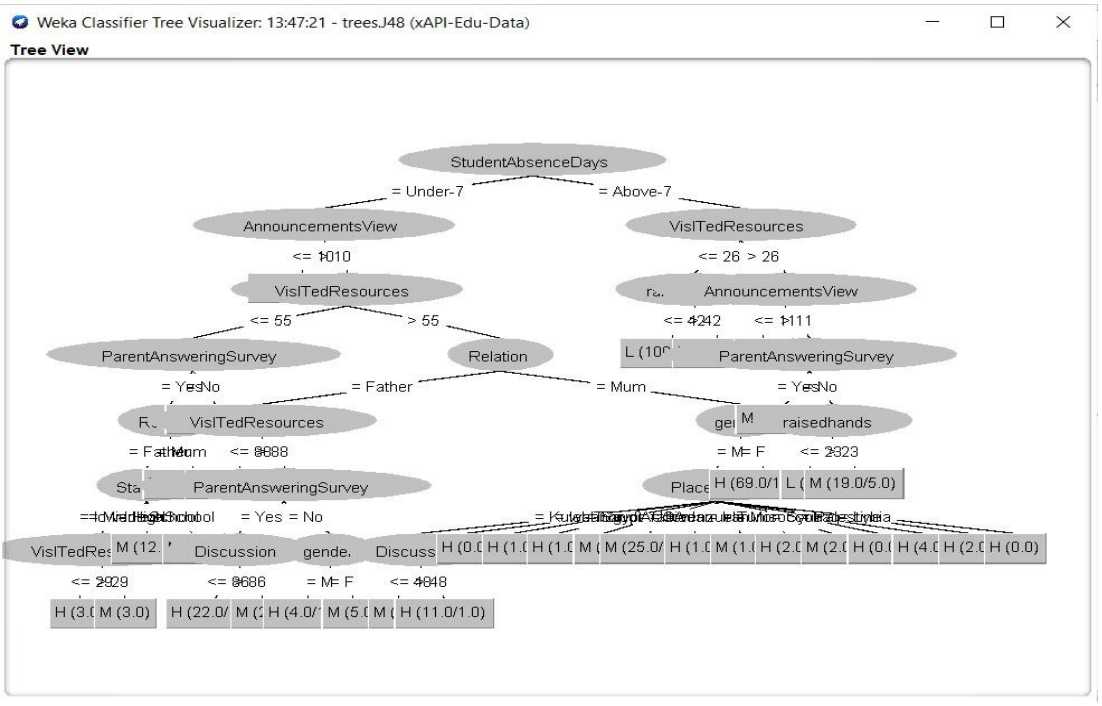


Fig viii – predicted decision Tree - for test dataset

Experiment results show the efficiency and effectiveness of the J48 algorithm in predicting the overall future result based on both academic results and outputs of other activities. The model also can improve the efficiency of the high ranked result achieving, keeping ,maintaining and improving position, job-position retrieving and evidently promote retrieval precision.

	TP rate	FP rate	Precision	Recall	F-measure	MCC	ROC-Area	PRC-Area	Class
	0.872	0.18	0.790	0.872	0.829	0.685	0.899	0855	M
	0.843	0.025	0.922	0.843	0.881	0.842	0.972	0.899	L
	0.796	0.053	0.863	0.796	0.828	0.761	0.950	0.867	H
Weighted avg	0.842	0.103	0.846	0.842	0.842	0.749	0.933	0.870	

Fig ix :- Statistical Accuracy

In fig ix - the ROC Area measurement is approximately greater than 0.89 for all classes that means the classification. The process succeeded in training the set. Thus, the predicted instances are similar to the training set, this proves the suggested classification model. The J48 algorithm can be used extracting and retrieving information to appear unseen information. The extracted information can be used to achieve
the quality in many fields.

Here in the above after implementation , result and analysis part it is clear that we can predict the future result based on previous results and according to our requirement we can choose , edit and modify attributes and instances to get better results. In the whole thing we discussed , analyzing the results comes from weak software though it is predicting the value comes from the same dataset as our student database management system to prove that through the process we have discussed above can really predict and get a close value in reality . but we can't not be able to add this feature to our student database management system directly. And that's why we use weka software . This is our ongoing project, so in future we will directly add this feature to predict the result inside our student database management system and establish a full Student DBMS whose feature are given below :

Future work and model brief :

1. Add the prediction model directly in the student database system.
2. We will add the student interface and faculty interface to our model where students and faculty can see their details and edit documents by login with their ID and password.
3. Students will be able to see the results of them and the future result prediction and also job-prediction based on their result and activities .
4. Admin and faculty will be able to see the individual student progression, result prediction , overall batch result prediction and according to that they can take necessary steps within time .
5. We will try to add a new feature where an institute can compare its overall points which is based on their academic performance and activities with previous year and give a predicted result for upcoming years .
6. We want to improved our home page, as it is the main thing which attracts all user
7. According to the requirements we will be developing this system because there is always something new that also is growing day by day.
8. Just not only in the student database management system , we want to use this classification method and develop the algorithm to use it in different types of non-commercial, commercial , business or any type of software where the future

prediction based on current situation will be beneficial for thinking and taking necessary steps.

CONCLUSION:

Simplicity is never simple. As we have seen in this project, the process of creating a user friendly and straightforward platform that facilitates the administrator's job is one filled with complexity. From understanding user requirements to system design and finally system prototype and finalization, every step requires in-depth understanding and commitment towards achieving the objectives of the project. In this paper, the student database is analyzed using J48 classification algorithm and placement related information is predicted based on academic results. The various data mining algorithms that can support the education system via generating valuable information are discussed. Data mining techniques can be very helpful in areas like prediction of academic results, student's placement and to improve students' academic results in higher institutions. The Data mining techniques also used in analyzing the student's academic results in different aspects as well as predicting the reason. Once data retrieved from the relevant resources, the classification algorithms can be applied to categorize the data. The feature selection of attributes from large data sets plays an important role in the efficiency of algorithms.

Although the student database management module is not fully integrated to the system and used in real time, the system prototype demonstrates easy navigation and data is stored in a systematic way. Overall, efficiency has improved and work processes simplified. Although all the objectives have been met, the system still has room for improvement. The system is robust and flexible enough for future upgrade using advanced technology and devices.

REFERENCES:

1. Wisdom Kwami Takramah, Wisdom Kwasi Atiwoto." Student Database System for Higher Education: A Case Study at School of Public Health", University of Ghana. American Journal of Software Engineering and Applications. Vol. 4, No. 2, 2015, pp. 23-34.

2. Vivienne V. Forrester. "School Management Information System: Challenges To Educational Decision- Making In The Big Data Era". International Journal on Integrating Technology in Education (IJITE) Vol.8, No.1, March 2019.

3. Shakeel Ahmad Dar. "Development of a Student Database Management System for the Polytechnic sector", International Journal of Advanced Engineering and Research Development. Vol 5, Issue 01, Jan -2018

4.Yu, Chong Ho, et al. "A data mining approach for identifying predictors of student retention from sophomore to junior year." Journal of Data Science 8.2 (2010): 307-325. 5. Mohammad Kamalal-Din Abbas."Students Fingerprint Attendance System for Faculty of Engineering using PHP & SQL" . Dept. EEE, University of Khartoum, Khartoum,Sudan, 2009.

5.Ahmed, Abeer Badr El Din, and Ibrahim Sayed Elaraby. "Data Mining: A Prediction for Student's Performance UsingClassification Method." World Journal of Computer Application and Technology 2.2 (2014): 43-47.

6.Jasser, Muhammed Basheer, et al. "Mining Students' Characteristics and Effects on University Preference Choice: A Case Study of Applied Marketing in Higher Education." International Journal of Computer Applications 67.21 (2013): 1-5.