OM SRI SAI RAM

SRI SATHYA SAI EDUCATIONAL TRUST (KARNATAKA)

[](https://www.bing.com/images/search?view=detailV2&ccid=pG3KCgKa&id=4CDAA32332B2681BD6ACDAC844858B263C88846C&thid=OIP.pG3KCgKaicd96K47sh_FzgAAAA&mediaurl=http://scriptures.ru/sarvadh.gif&exph=210&expw=210&q=sarva+dharma+logo&simid=607987735829741713&selectedIndex=87)

Sri. Sathya Sai Institute for Women

4TH Cross, Saptapur, Dharwad – 580 001

Report on

School Tool

(Live Project)

Submitted By

**Aishwarya Belligatti Nagaveeni Barker**

**Acknowledgement**

We offer our gratitude to all the people without whose help the project would not be possible.

We would like to extend our sincere and profound gratitude to **Mr. Kumar Panchappa**, Principal of our college for providing invaluable guidance, comments and suggestion throughout the course of the project.

We would also like to specially thank our project guide **Smt. Renuka Kongi** for helping us and constantly guiding and motivating us to work hard. Furthermore, we thankfully acknowledge the encouragement received from other lectures of BCA department of my institution. **Smt. Indira Melagmalagi, Smt. Uma Joshi, Smt. Nanda Telkar** for giving us support.

We are immensely grateful to our project guide Mr. Peter Sunny for his valuable encouragement, knowledge and co-operation.

Finally we would like to express our gratitude to our parents, friends for their moral support and Inspiration during the project.

**Contents**

1. **Introduction**

1.1 Objective

1.2 Existing System

1.3 Proposed System

1. **Requirements**

2.1 Hardware requirements

2.2 Software requirements

1. **Analysis Document**

3.1 Data flow diagram (DFD)

3.2 Use Case diagram

1. **Design Document**

4.1 Database Design

4.2 Interface Design

1. **Coding**
2. **Conclusion**
3. **Bibliography**

**1 - Introduction**

The school tool is a live Web-Application based system which will be used as a platform for interaction between student, teachers and parents. While the main objective of this project is to automate the work. The computerization is done so that details of students, teachers, events will be stored in the system. The chance of deletion or duplicating of data will be almost reduced. While doing the automation to the system will reduce the time for storing and maintaining data.

Using School Tool will automate academic processes to save time and reduce staff workload. Students no more have to stand for hours in the queue to pay fees. Simplify registration and fee collections with online forms. Dashboard reports and intelligent analytics are useful indicators for educators to examine attendance, assignments, grades, etc. and predict student outcomes.

**1.1 Objectives**

Now a day’s computer plays a very important role in any formal organization, because computer works faster and accurate than human. The purpose of developing the **SCHOOL TOOL** for any organization is that to make easy for management to store a large number of data/reports into the computer. Following are the few objectives of School Tool System:

* Fees management
* Admission Management
* Student performance monitoring
* Organise and simplify everyday tasks
* Centralised data and easy access to all
* Attendance management
* Expenditure management

**1.2 Existing System**

Educational institutions are burdened by cumbersome paperwork and manual processes, and they find it difficult to maintain records on attendance, fees, admissions, transport etc., and track the information they need.

There is chance of deletion or duplicating of data in existing manual system. Existing system consumes lot of time and labor work for maintaining records and details.

**1.3 Proposed System**

This school management system having Two main management tools, Student, Teachers, management system Student management has some sub categories registration, attendants, Grade management and Report generating. Reports are for individual performance, overall performance, for one subject etc. Teachers management attendants, add, update, delete and report generation.

**Teachers Management**

* Teachers management Dip can add, update, delete Teachers registration.
* Teachers can mark attendance every day.
* Teachers management Dip can generate, print ID card for Teachers
* Teachers management Dip can Add salary details can’t edit.
* Teachers management Dip can add, update, delete class teachers.
* Attendance detail’s report generated every teacher.
* Per teacher’s attendance report can generate monthly, yearly.

**Student Management**

* Student management Department can add, update, delete student registration.
* Class Teachers can mark attendance of all students of his class only.
* Teachers, students, parents can view students marks details.
* Student management Dip can add payment details but can’t edit.
* Exams result are generated by students’ grade and student by student.
* Attendance details report generated by class by class
* per class attendance report can generate monthly, yearly

\*\*\*\*\*

**2 - Requirements**

**2.1 Hardware requirements**

Processor: Intel Pentium or More

RAM: 512 MB

Hard Disk: 50GB

**2.2 Software requirements:**

**Front End**

* HTML -5
* CSS -3

**Client-Side scripting language**

* Java script

**Frontend Frame Work**

* Bootstrap
* Material UI

**Back End**

* PHP

**Database**

* MySQL

**Local Server**

* XAMP

**Text Editor**

* Sublime

**HTML**

HTML (Hypertext Markup Language) is the set of symbols or codes used to display the contents of a file. Each markup symbol is called as an element usually people also refer to it as a tag. Some symbols are in pairs that indicate to some display and to begin and to end. The content of the page should be written in body tag and in the html structure the innermost tag will close first and followed by next.HTML was developed by Berners-Lee and is known as the father of HTML.

The html language is used to construct the forms and which acts as the channels through which data can be inserted. Consider the following instance where a form is created using this markup language. Let us design a form for login page with two labels, two text box and a submit button.

**CSS**

Cascading Style Sheets, fondly referred to as CSS, is a simple design language intended to simplify the process of making web pages presentable.

CSS handles the look and feel part of a web page. Using CSS, you can control the text, the style of fonts, the spacing between paragraph, how columns are sized and laid out, what background images or colors are used, layout designs, variations in display for different devices and screen sizes as well as a variety of other effects.

CSS is easy to learn and understand but it provides powerful control over the presentation of HTML document. Most commonly, CSS is combined with the markup languages HTML or XHTML.

**Java Script**

It is a programming language that can be included in forms to make them more interactive. Used for checking or modifying the inputs of forms, change images, open new windows and write dynamic page content. The main purpose to use java script is to make validation for the entered data in the form.

Brendan Eich was developer of Java script by, who works at Mozilla. The real name is Mocha, named by the founder of Netscape. In 1995 the name got changed to Live Script, in the same year on December, java script was replaced.

*The features of JS:*

Helps to detect user’s browser.

Helps user to have control over the browser.

Does simple manipulations on the client side.

Helps to handle dates and times.

Validating the users input.

**PHP**

Other name is "Hypertext Preprocessor”. A Web language that is HTML-embedded. Where, PHP code can be written along with the html. When a PHP page is accessed, the PHP code is read or "parsed" by the server. The outputs from the PHP functions are always returned as HTML code, and the browser reads it. PHP code is converted into HTML before the page is loaded therefore users cannot view the PHP code on a page. This make PHP pages safe to access databases and other vital information.

PHP development was started in 1994 by Rasmus Lerdorf and he is known as the father of PHP. In the following situation, the admin enters the details of new subject in the form.

Admin Enters the subject name, description and the no of hours and later clicks on the submit button. All the entered data should get inserted in the respective table. To make this thing possible the php code is used. In that, first the connection with the database is established. All the form data, that is name, description, and the no of hours entered is collected through post variable. Then using the suitable SQL query the data is inserted into the subject table.

Using the PHP code, the existing information about a particular subject can be changed and similarly if required the information can be deleted. In this way PHP code, helps to do the same activities on different data.

**My SQL**

For creating the database and the tables for the database it is used. This helps to store the vital information of the application for longer time and can be accessed after few months or years. This is very much useful to keep the details for any of the information related to the lecturer or students.

**3: Analysis Document**

**3.1 Data flow diagram (DFD)**

A **data flow diagram** (**DFD**) is a graphical representation of the "flow" of data through an information system, modeling its process aspects. A **DFD** is often used as a preliminary step to create an overview of the system, which can later be elaborated.

The DFD is a graphical technique that depicts the information flow and the transformation that are applies as data move from input to output. The DFD is also known as Bubble chart or data flow graphs or Context diagram.

The data flow diagram may be used to present a system or software at any level of abstraction. A functional system bubble with input and output data indicated by incoming and outgoing rows respectively.

*Data flow diagrams are constructed from four basic buildings blocks.*

* Process
* Data Flow
* Store
* Terminator

**The Process:**

Also, called bubble, function and transformation. Shows part of the system that transforms inputs to outputs. Represented graphically as a circle. Named with single word, phase, or Sentences.

**Data Flow:**

It is represented by and arrow line and name of the data is specified by the said of the line as label. This is used for data movement.

**Two data packers:**

Represented graphically by an arrow into out of service process. Describes movements of information in the Systems Data in motion. If flow shows direction. Double- headed arrow stands for dialogue- Convenient packaging of two data packers. Data flows can diverge or converge in a DFD.

**Data Store**

It is representing by one open end rectangle the data base used in the system are specified by this function.

**Source or Links:**

Identify System, processing transformations. Transactions concerned reading, validating and formatting input. Identify input transformations. Transformations concerned with validating and formatting inputs. Identify output transformations. Transformations concerned with and writing output. Group under the output.

**3.2 Use case diagram**

A UML use case diagram is the primary form of system or software requirements for a new software program underdeveloped. Use cases specify the expected behavior (what), and not the exact method of making it happen (how). So use case diagrams are usually referred to as behavior diagrams used to describe a set of actions (use cases) that some system or systems (subject) should or can perform in collaboration with one or more external users of the system (actors). Each use case should provide some observable and valuable result to the actors or other stakeholders of the system.

Use cases once specified can be denoted both textual and visual representation (i.e. use case diagram). A key concept of use case modeling is that it helps us design a system from the end user's perspective. It is an effective technique for communicating system behavior in the user's terms by specifying all externally visible system behavior.

*A use case diagram is usually simple. It does not show the detail of the use cases:*

It only summarizes some of the relationships between use cases, actors, and systems.

It does not show the order in which steps are performed to achieve the goals of each use case.

Use Case diagrams are constructed from four basic buildings blocks.

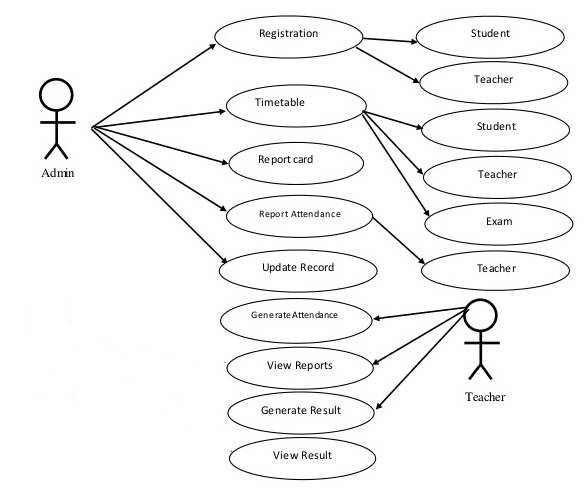
Actor

Use case

Communication link

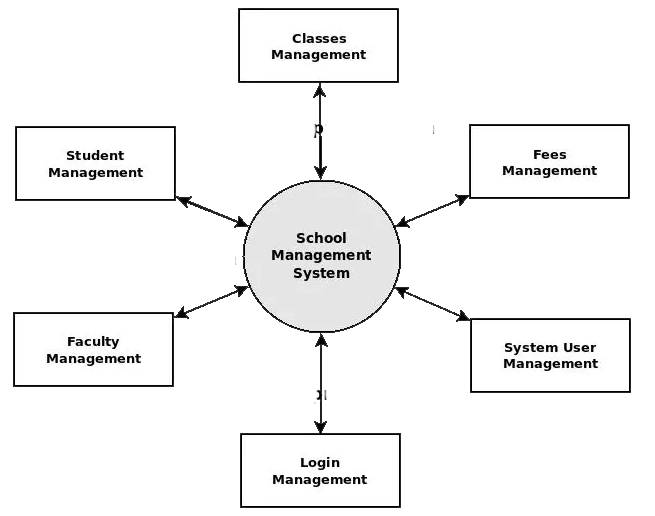
Boundary of system

**Use Case of School Tool**

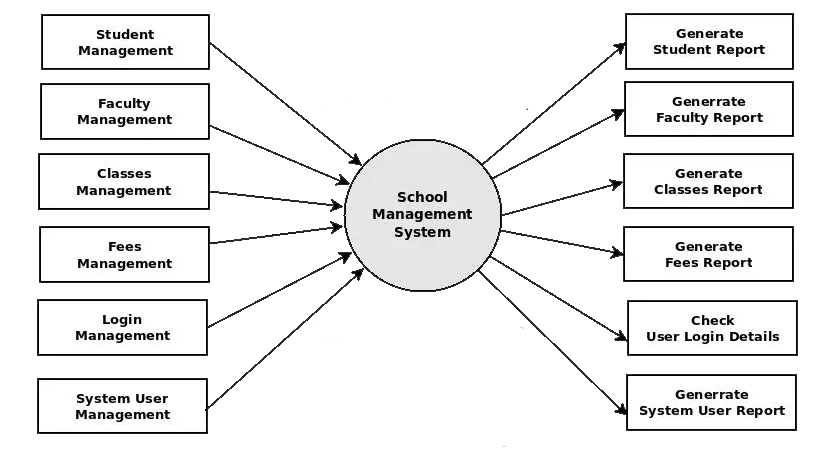


**Data Flow Diagram**

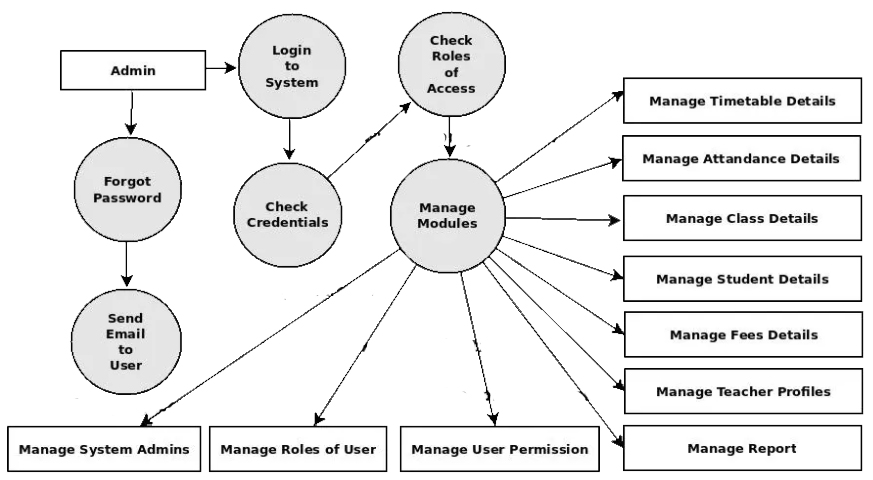
1. **Zero Level**



1. **First Level**



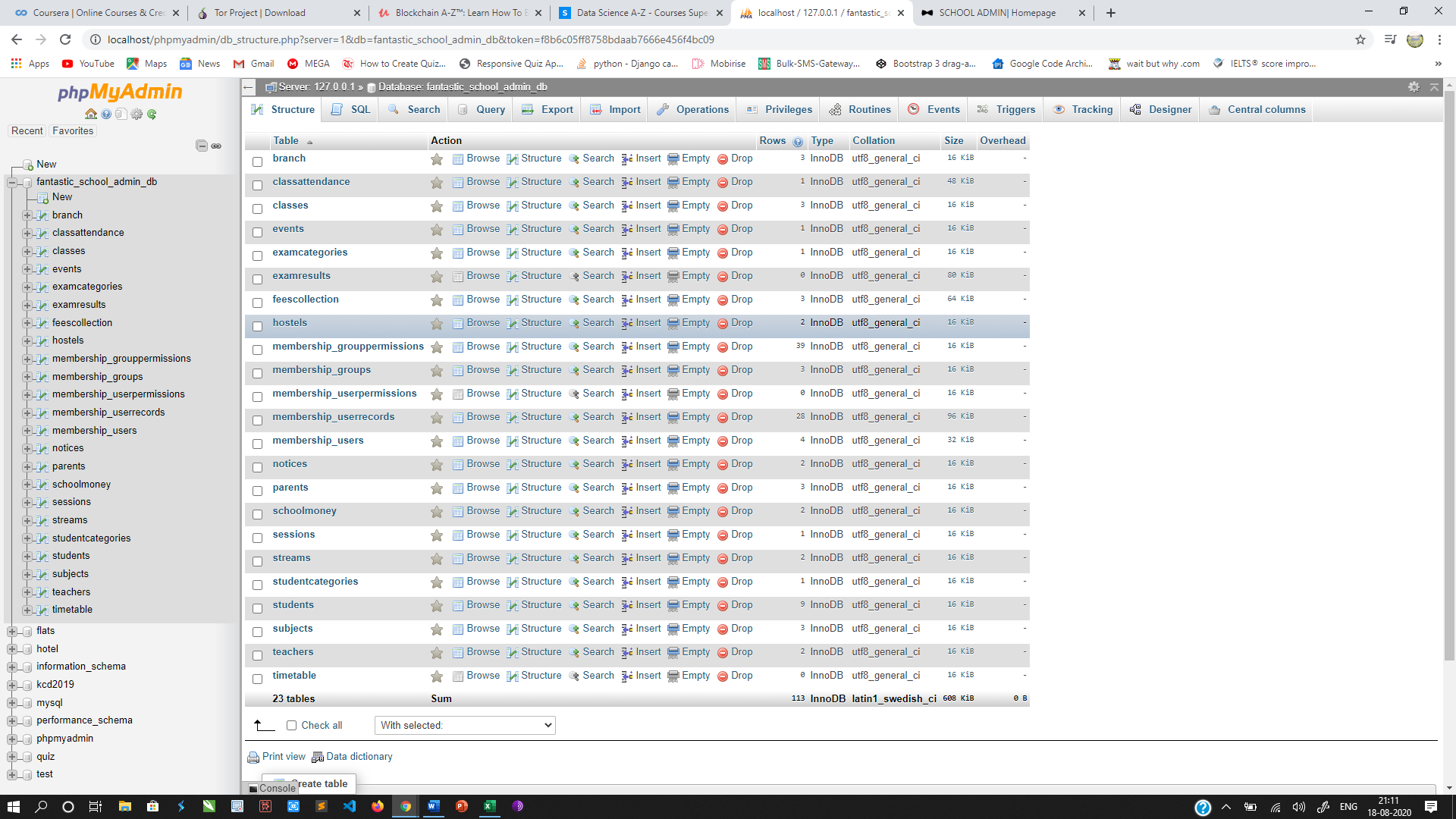
1. **Second Level**

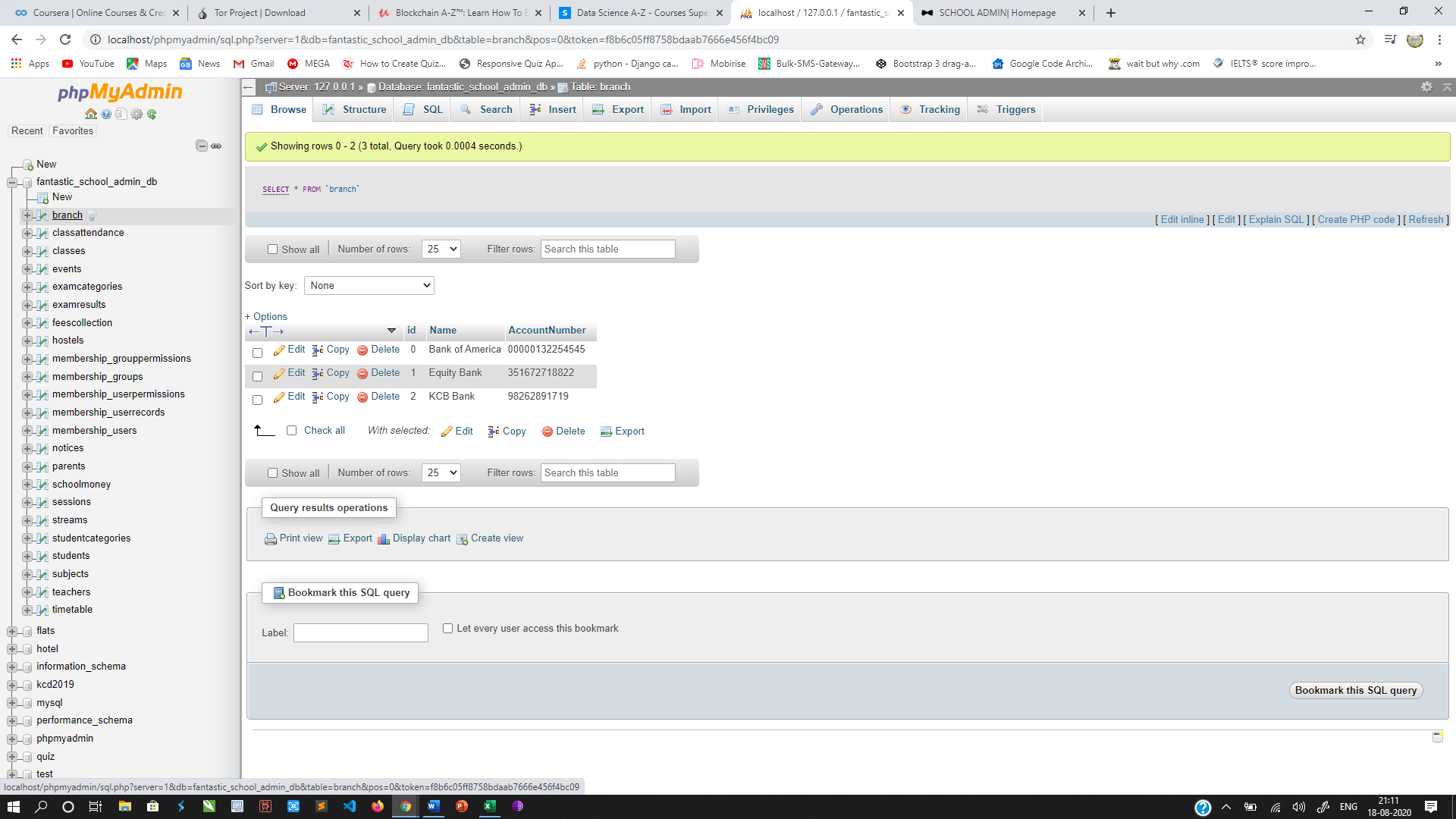


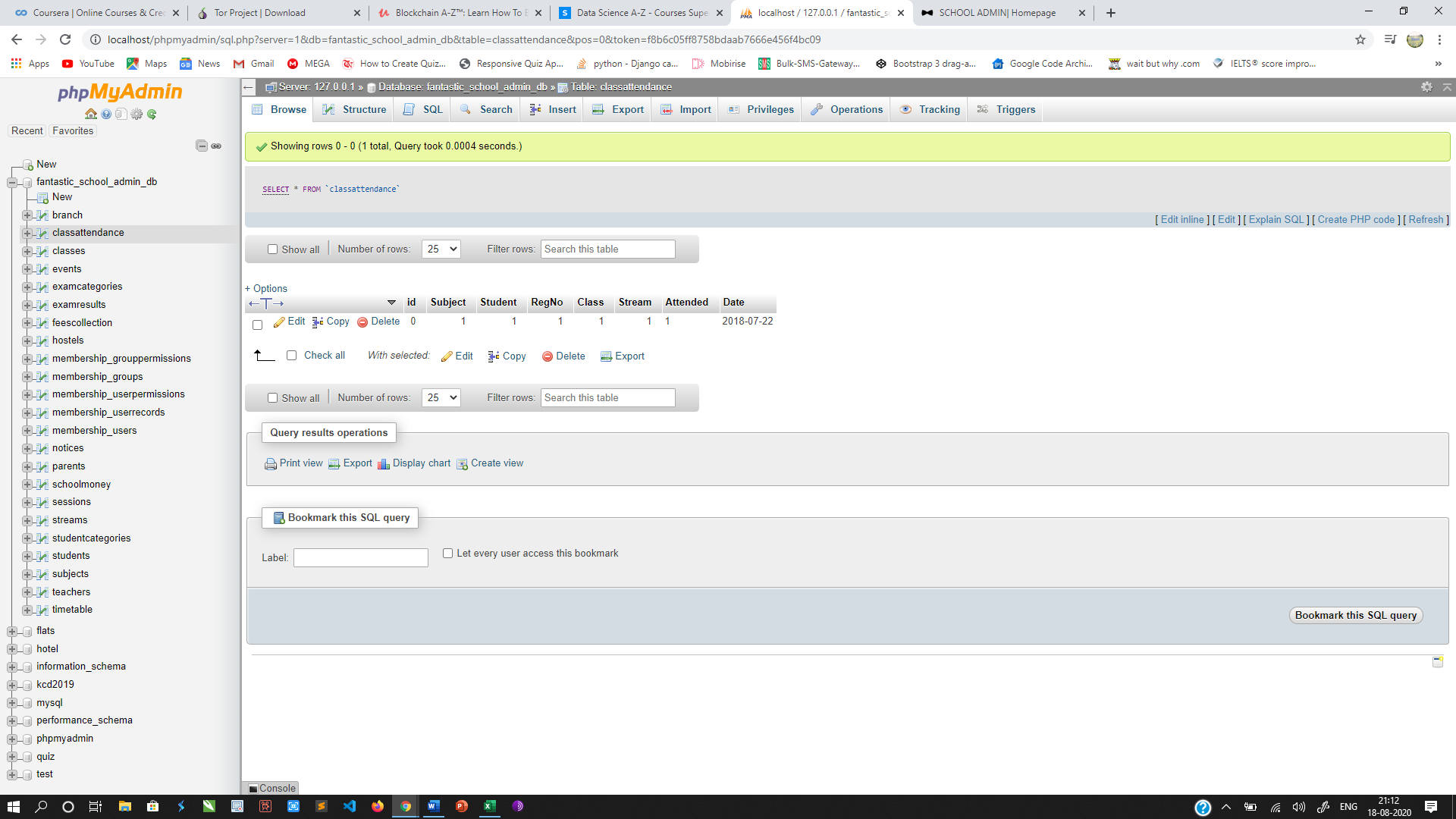
**4 Design Document**

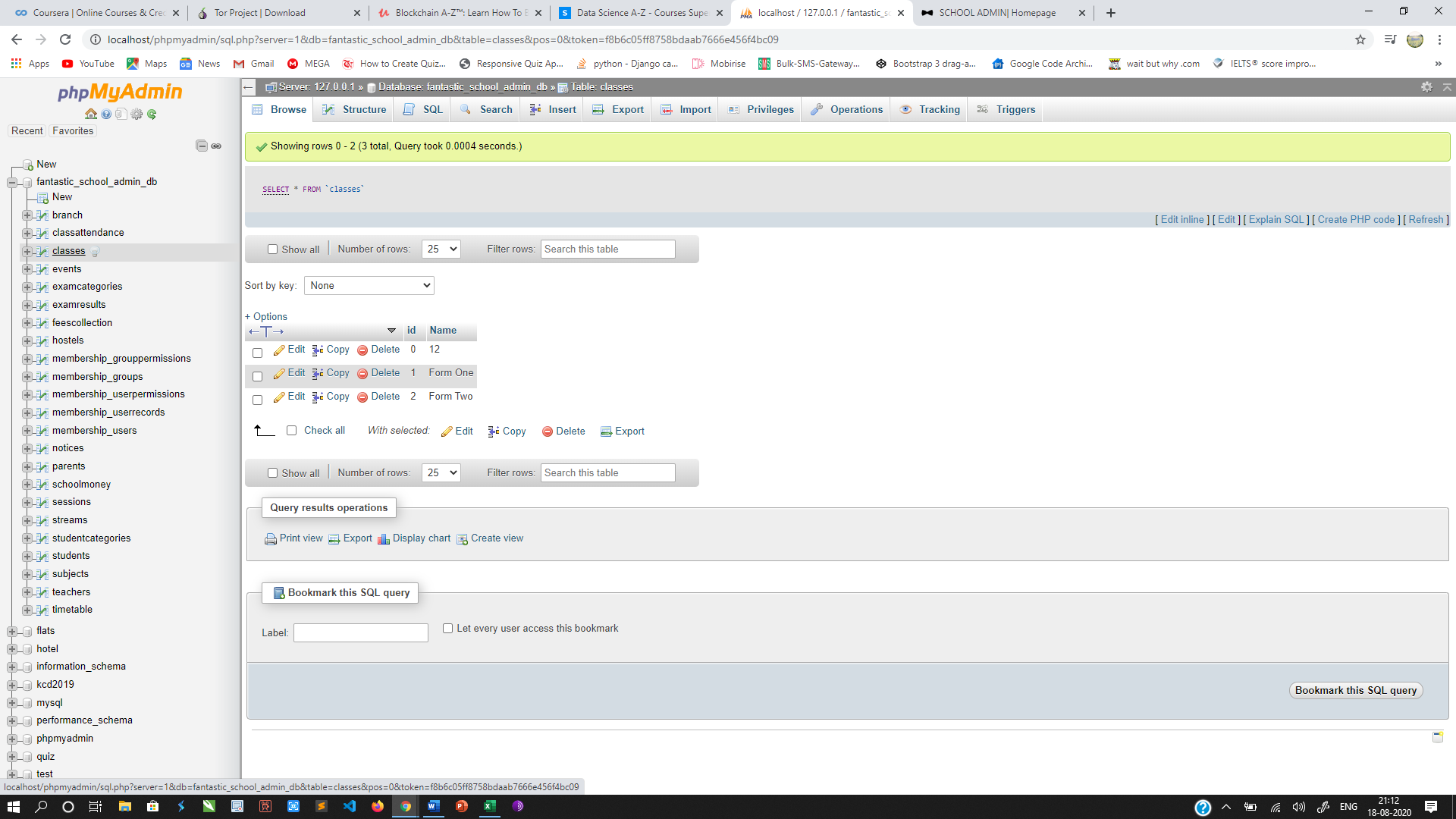
**4.1: Database Design**

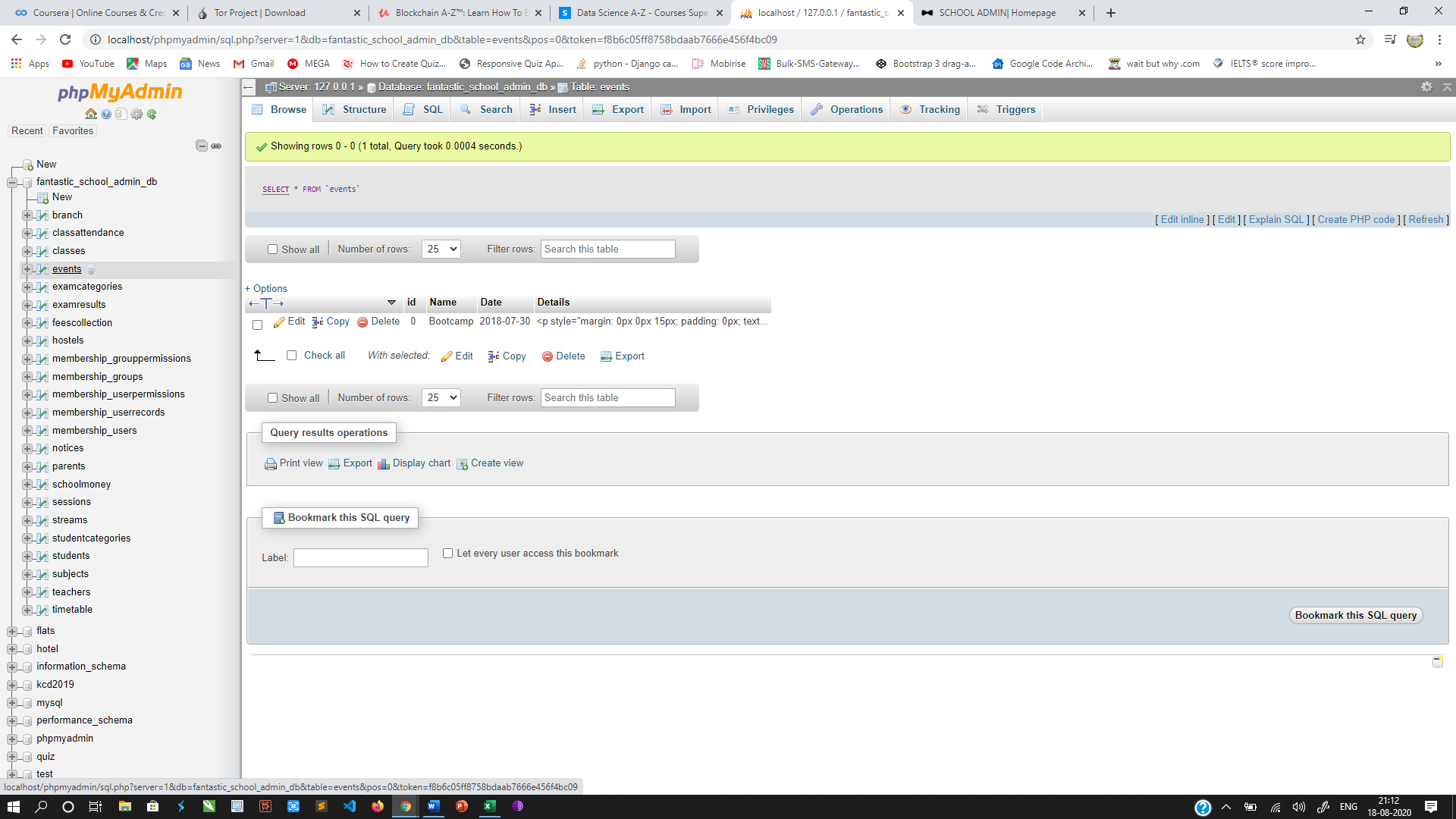
A database is a stored collection of interrelated data, organized on basis of relationship in the data rather than the convenience of storage structures. It enables sharing of data among various users as and when required. A database, in which data is stored in tables allowing relationship among tables and more efficient non-redundant data storage. For our project we have used MySQL for storing data.

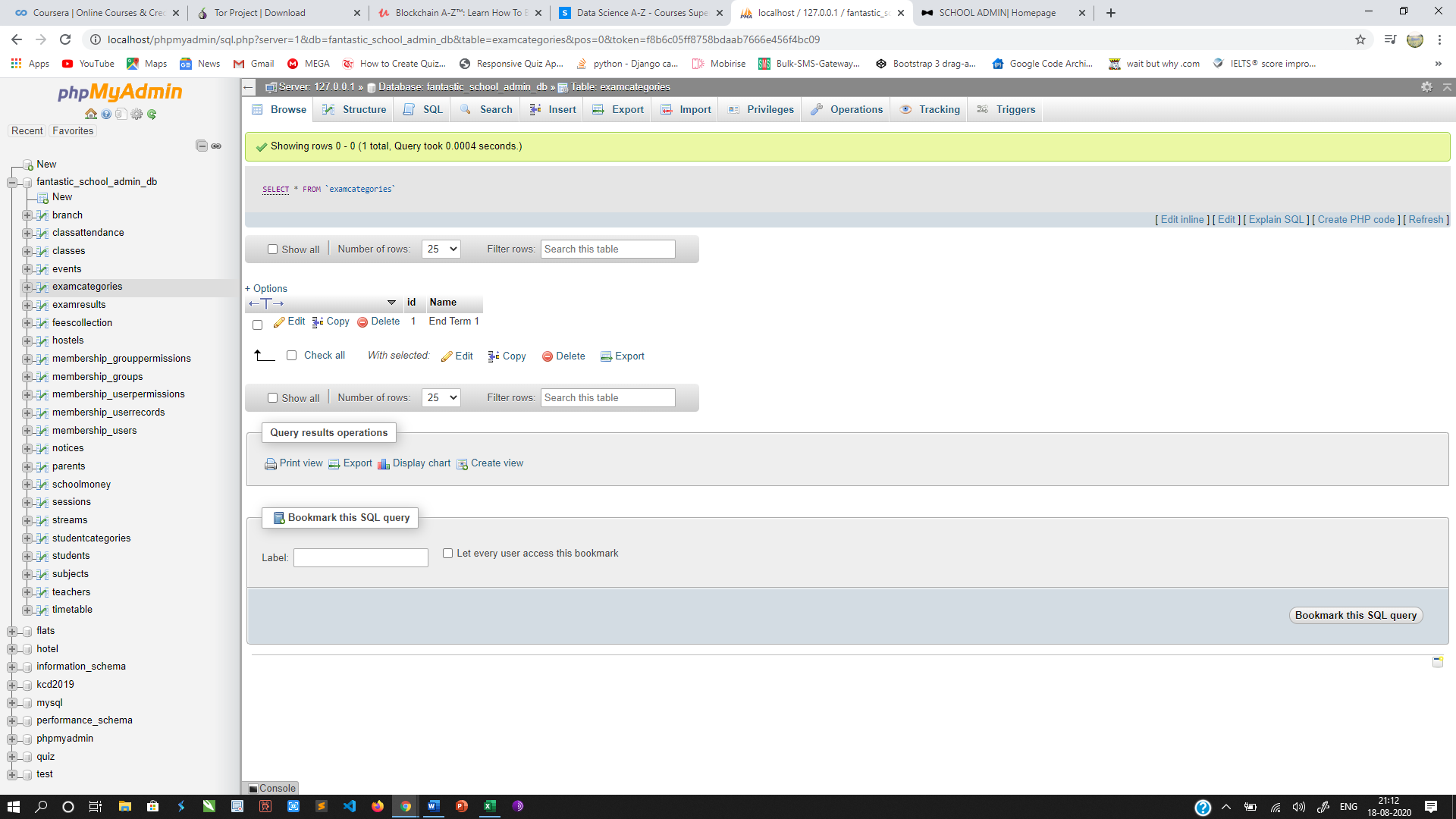


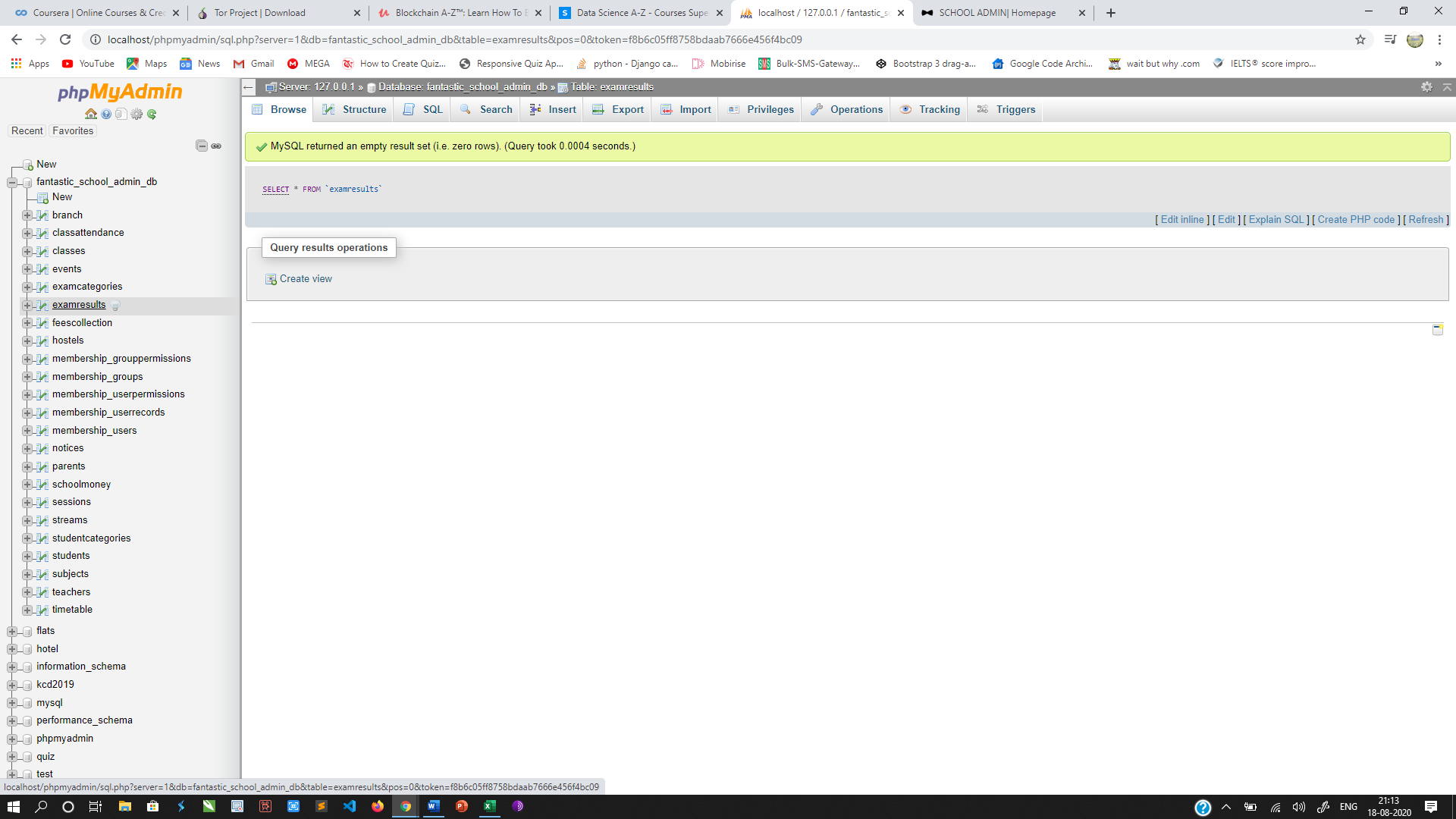


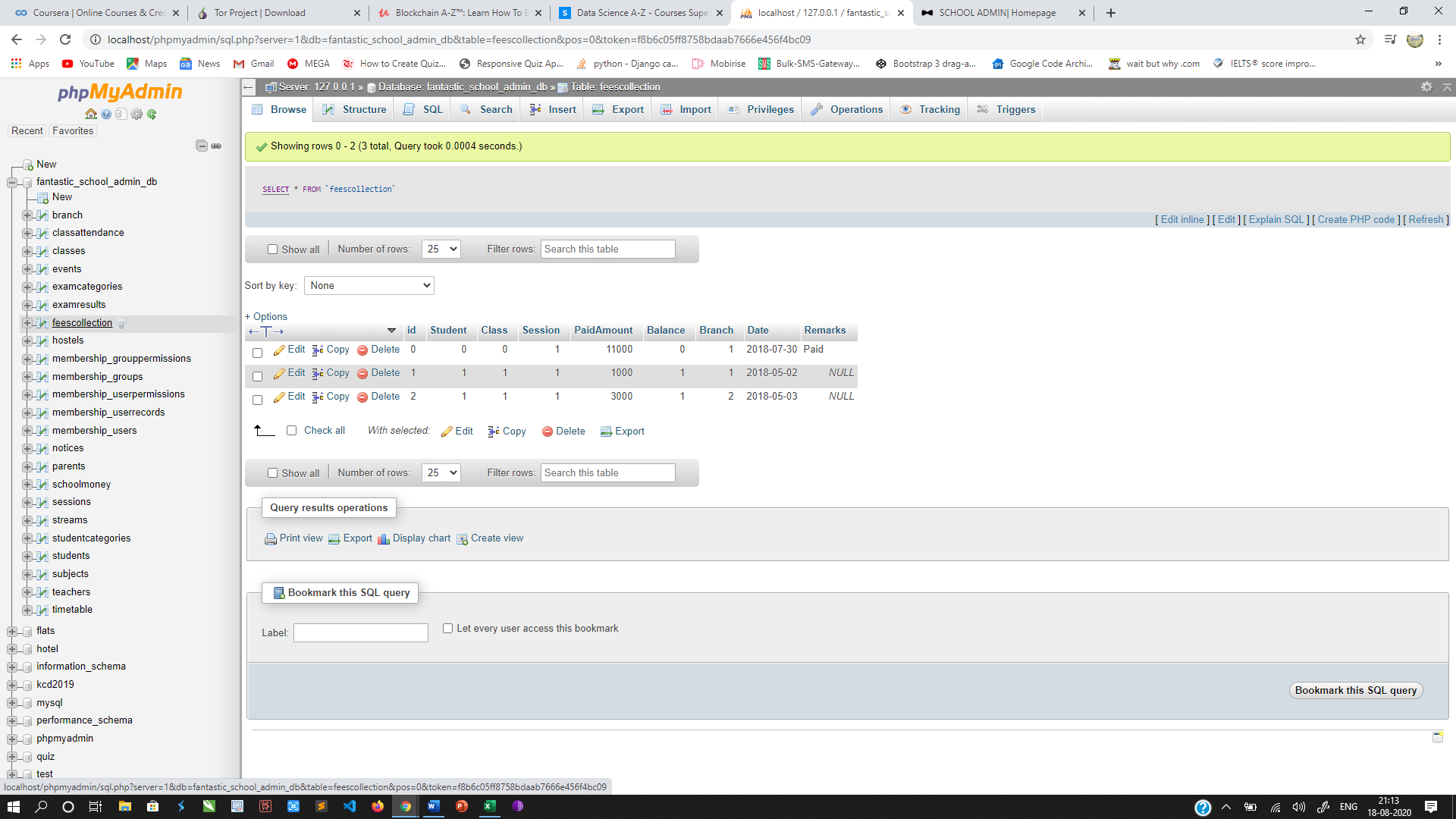


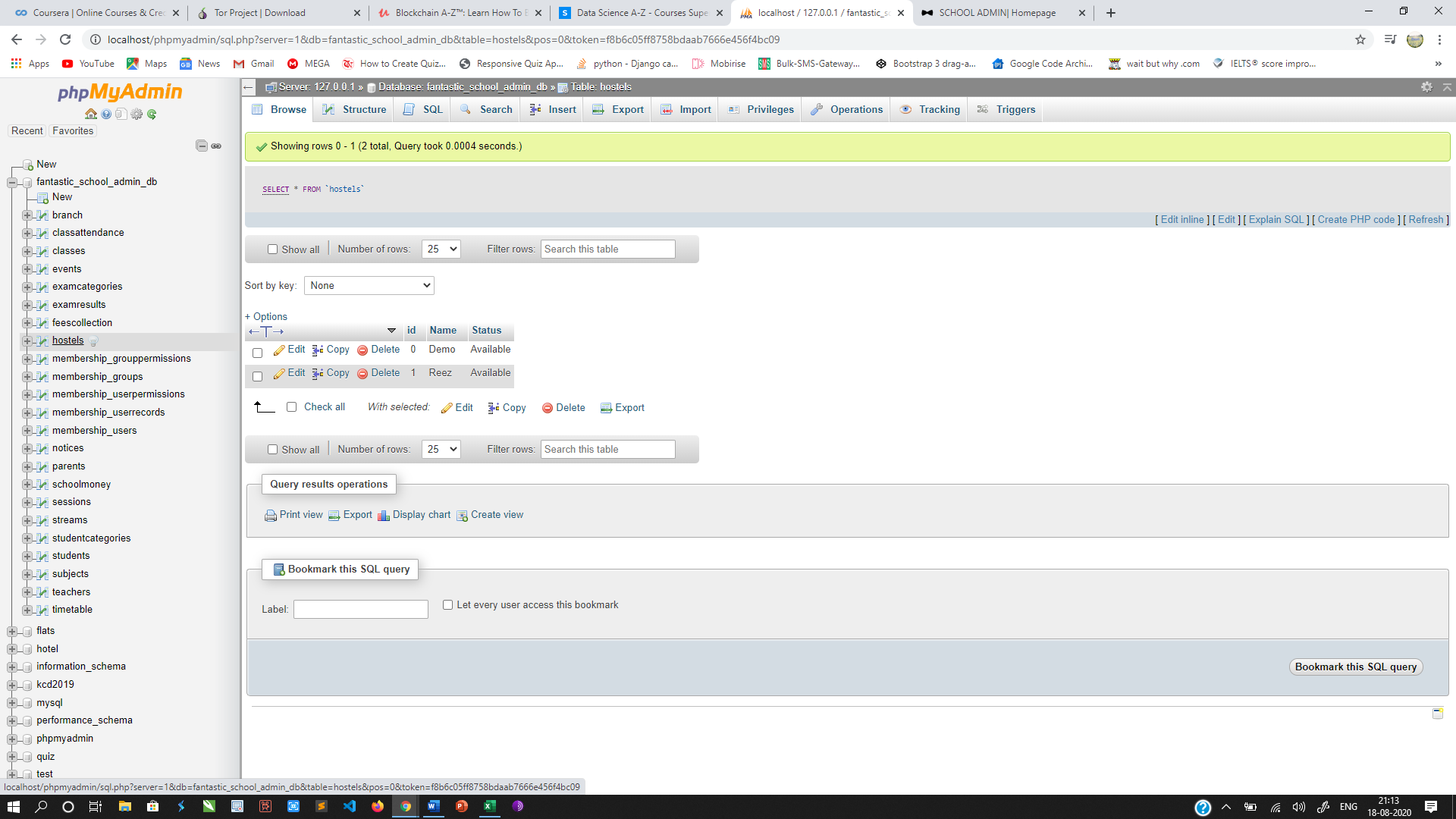


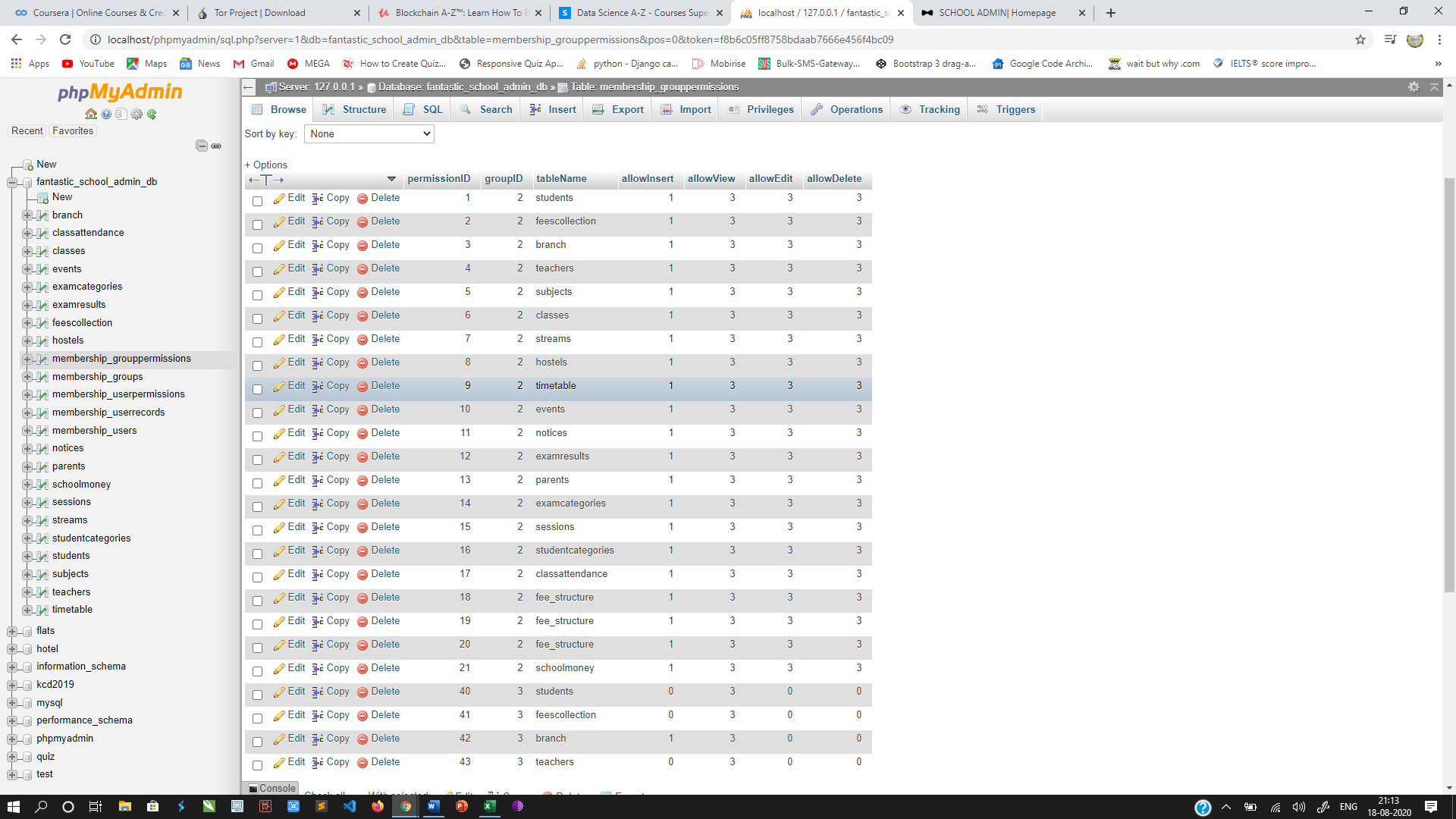


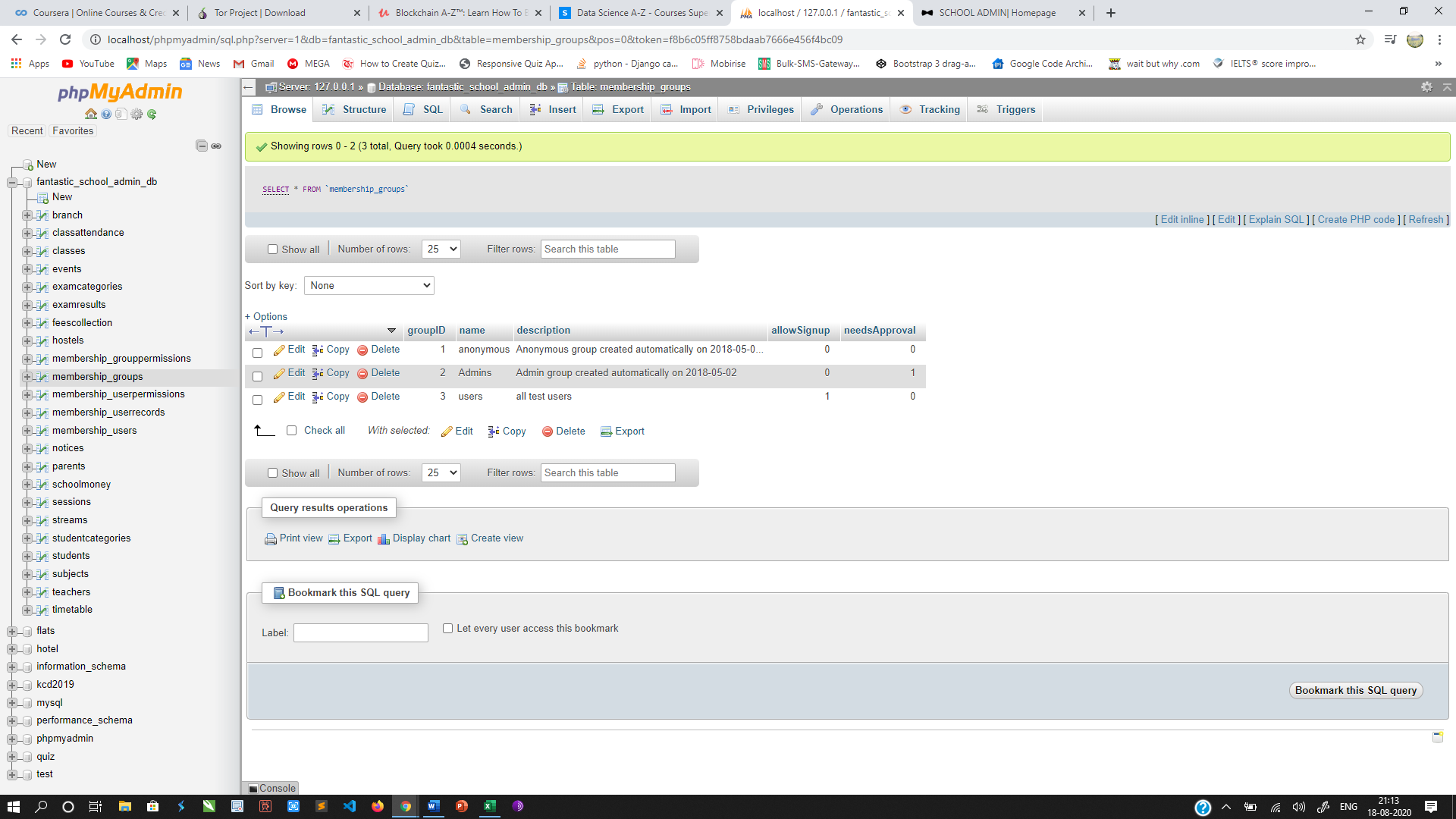


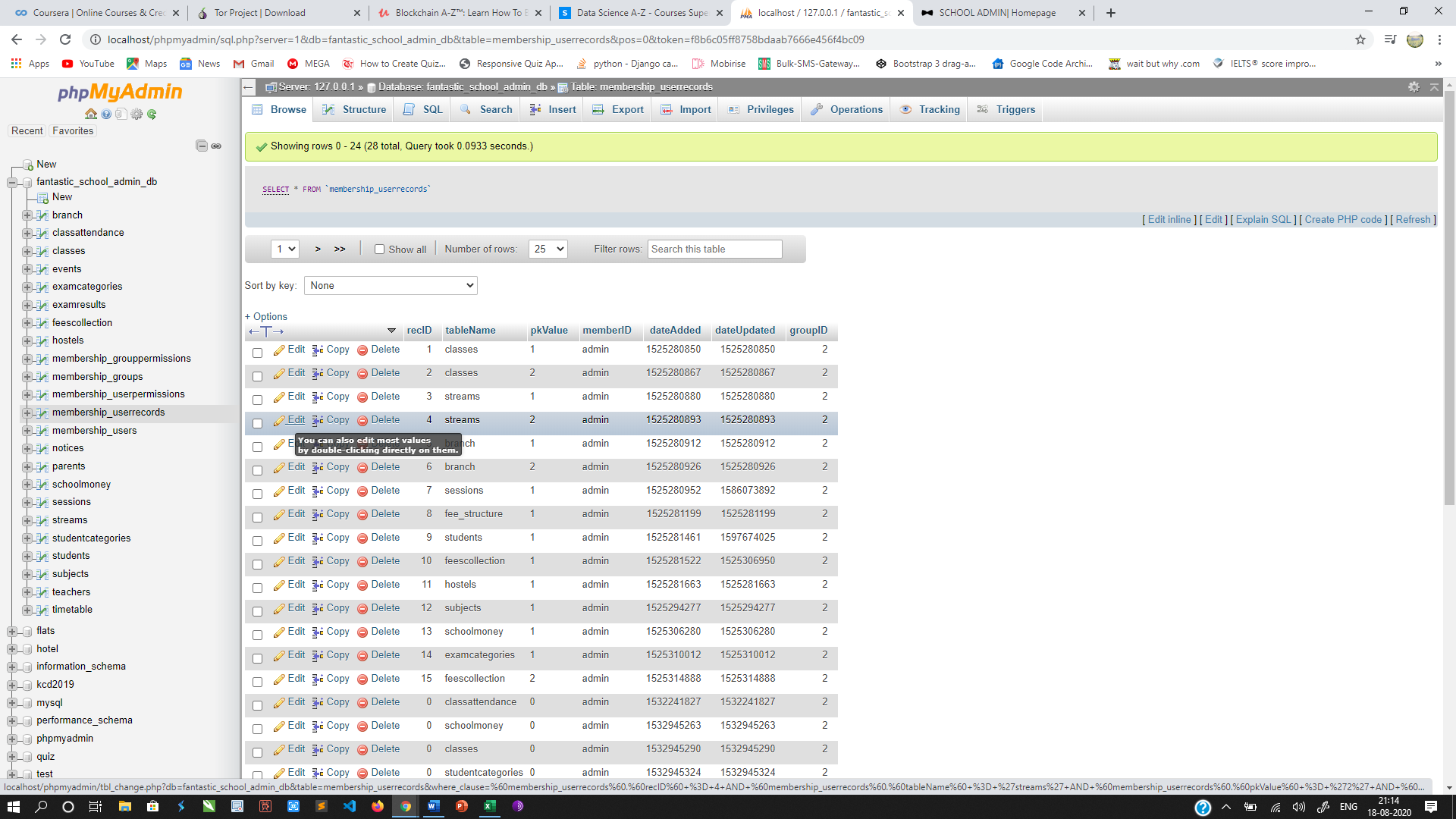


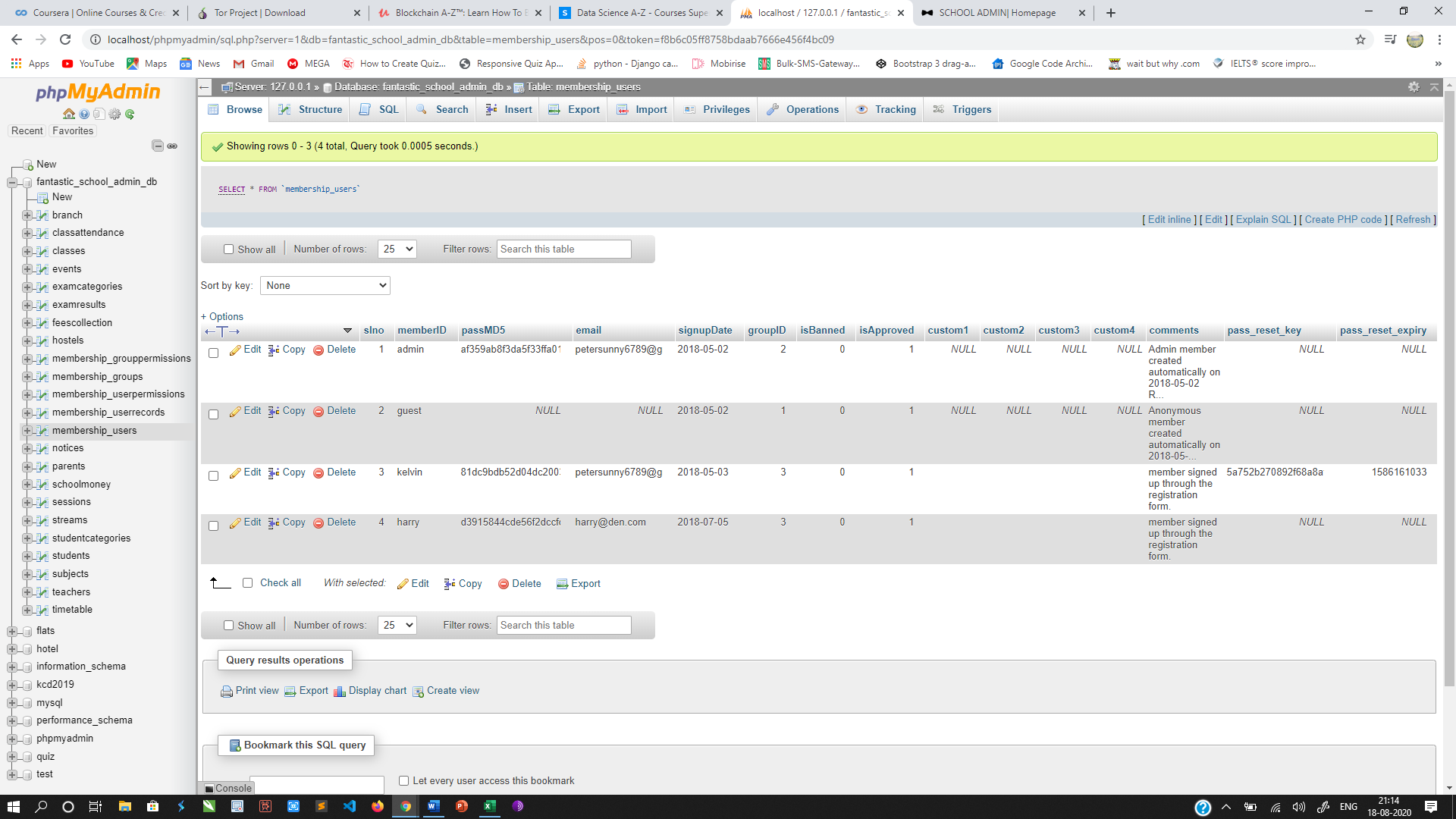


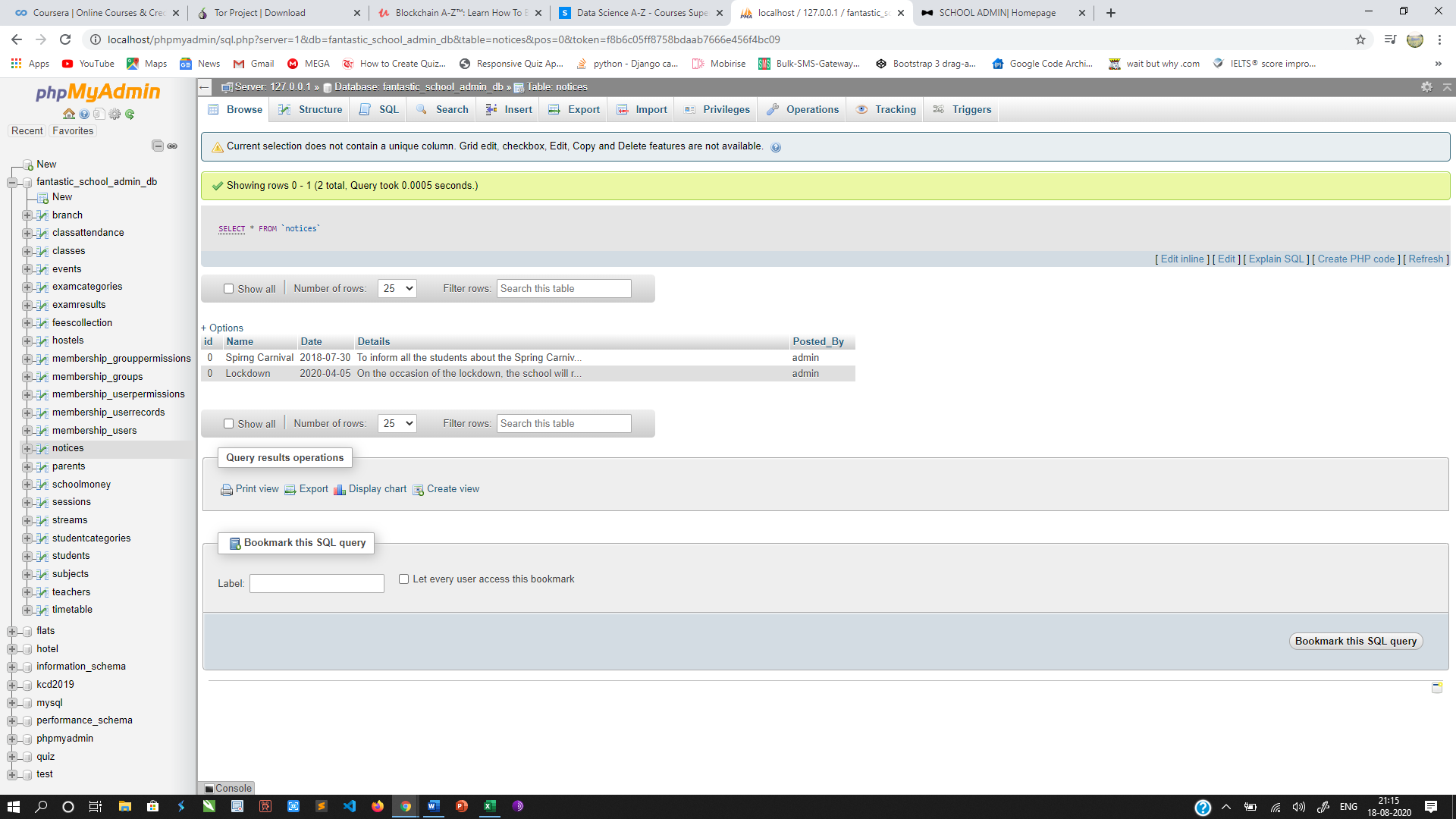


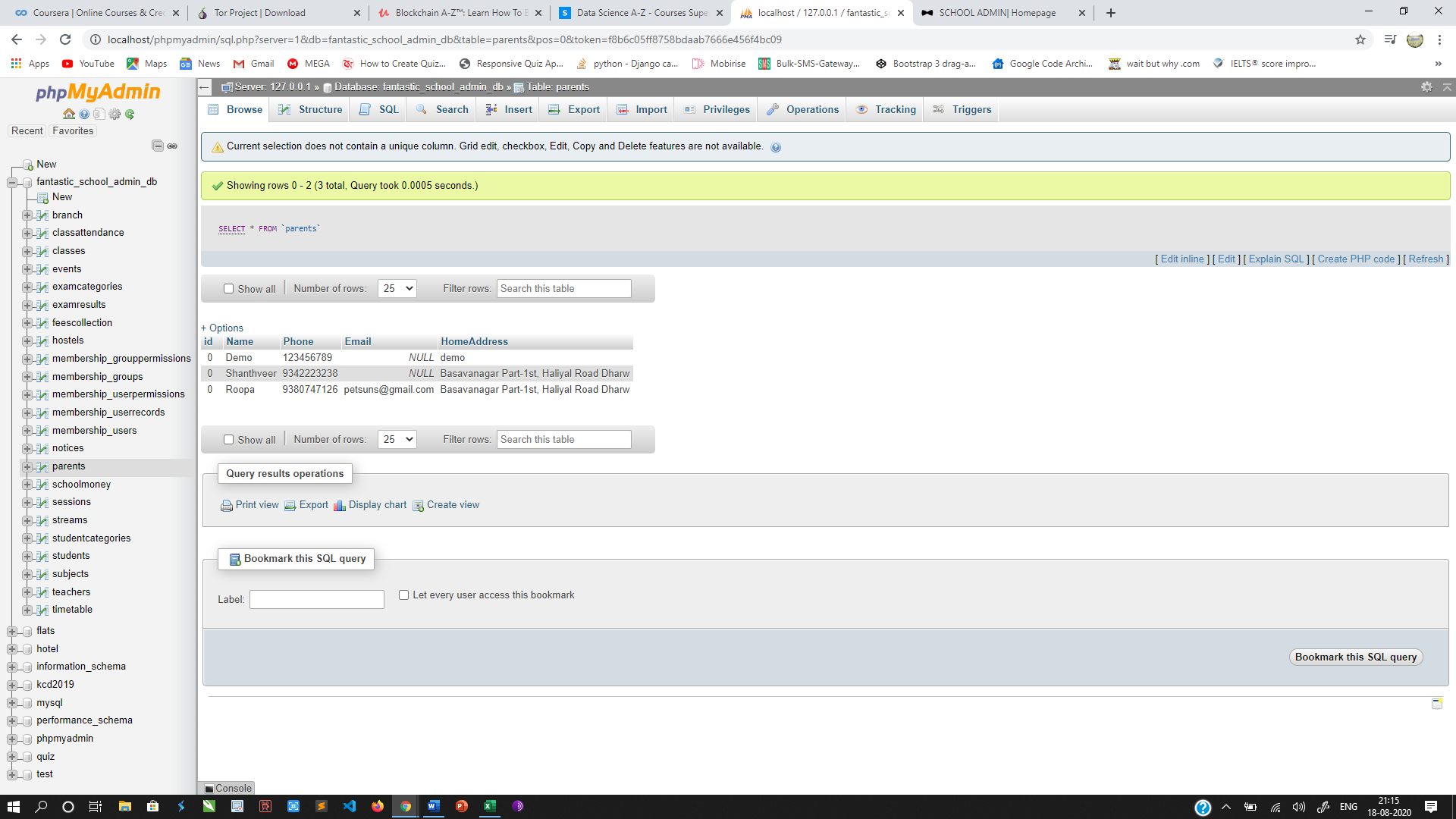


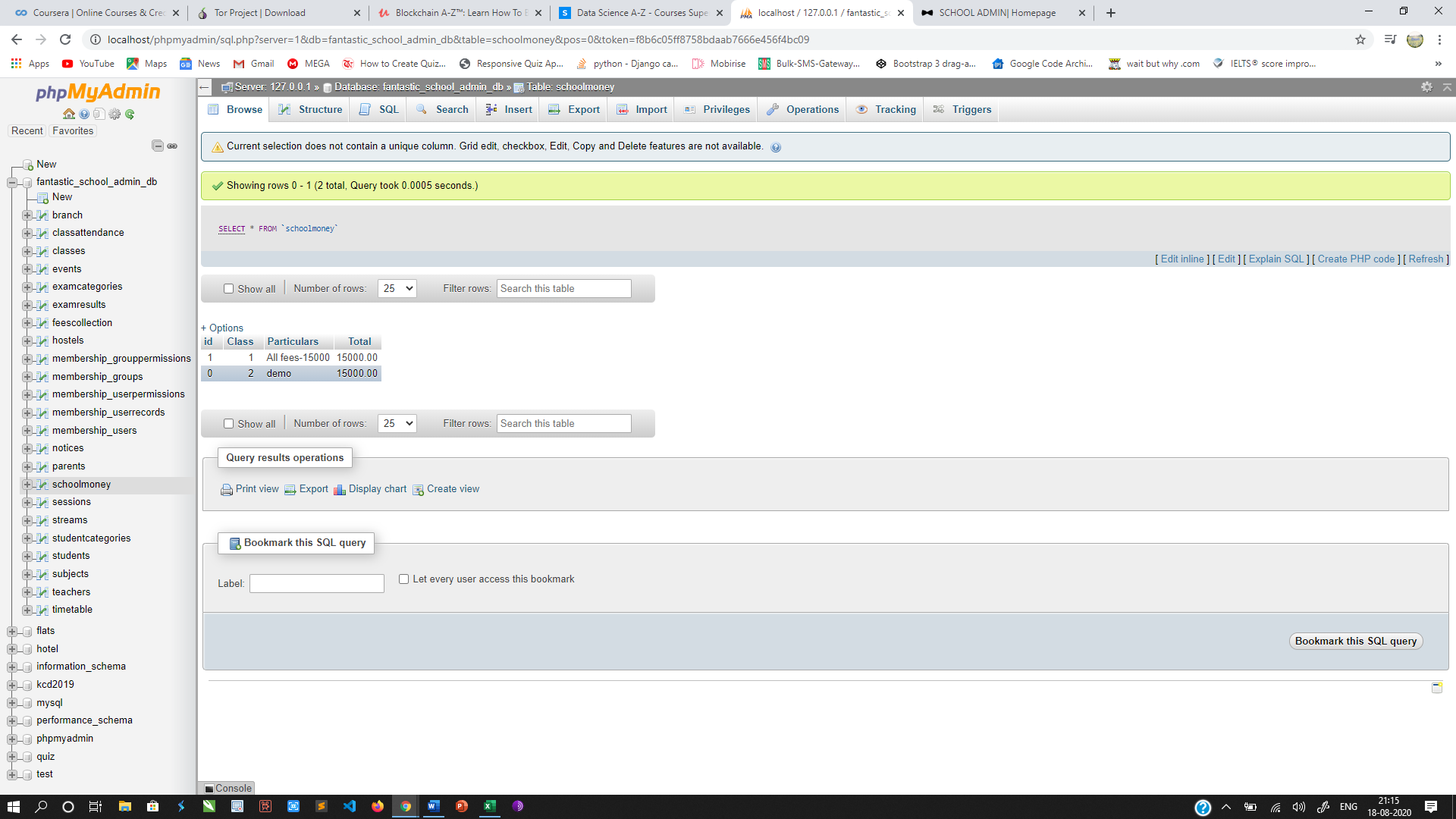


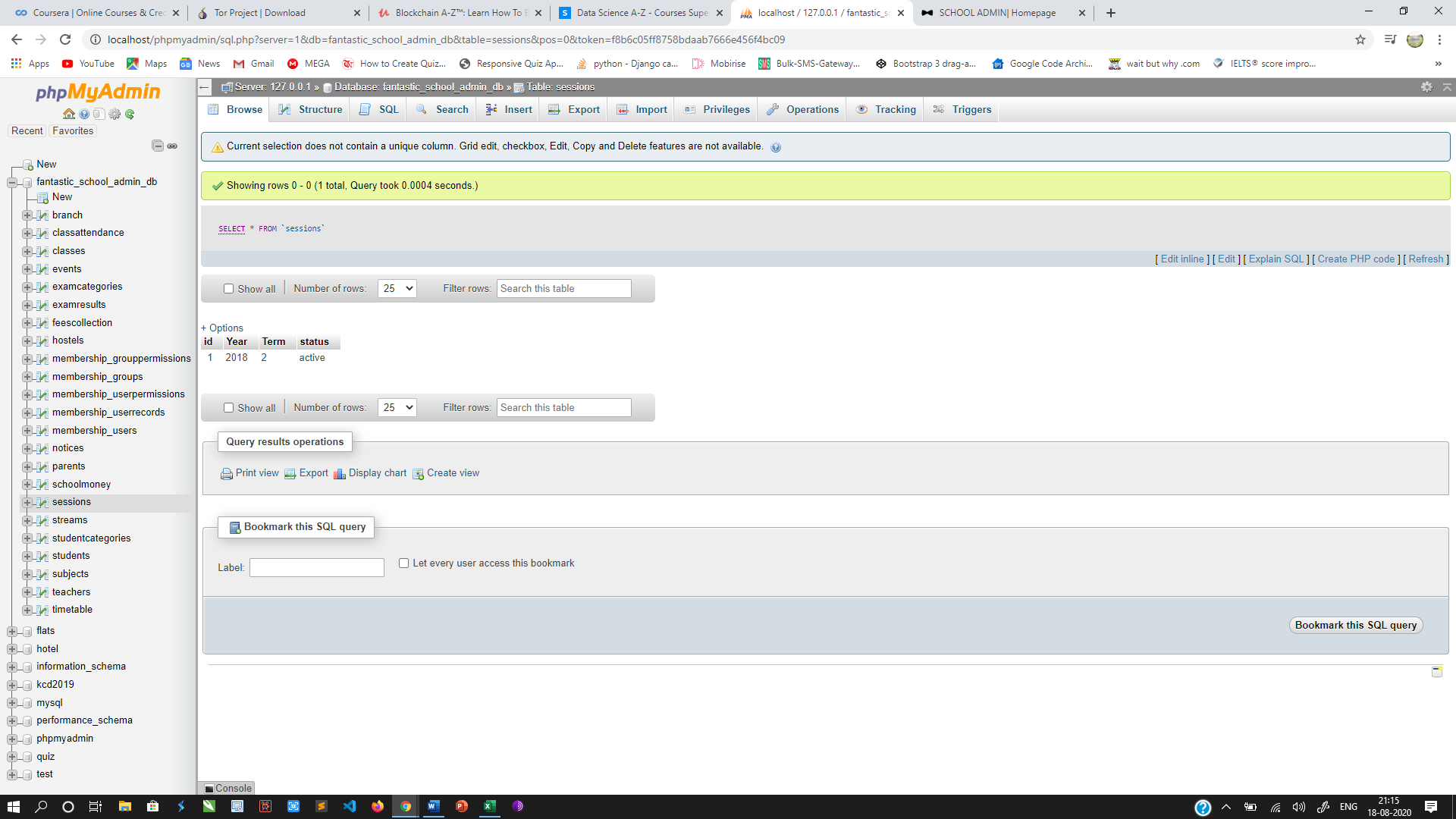


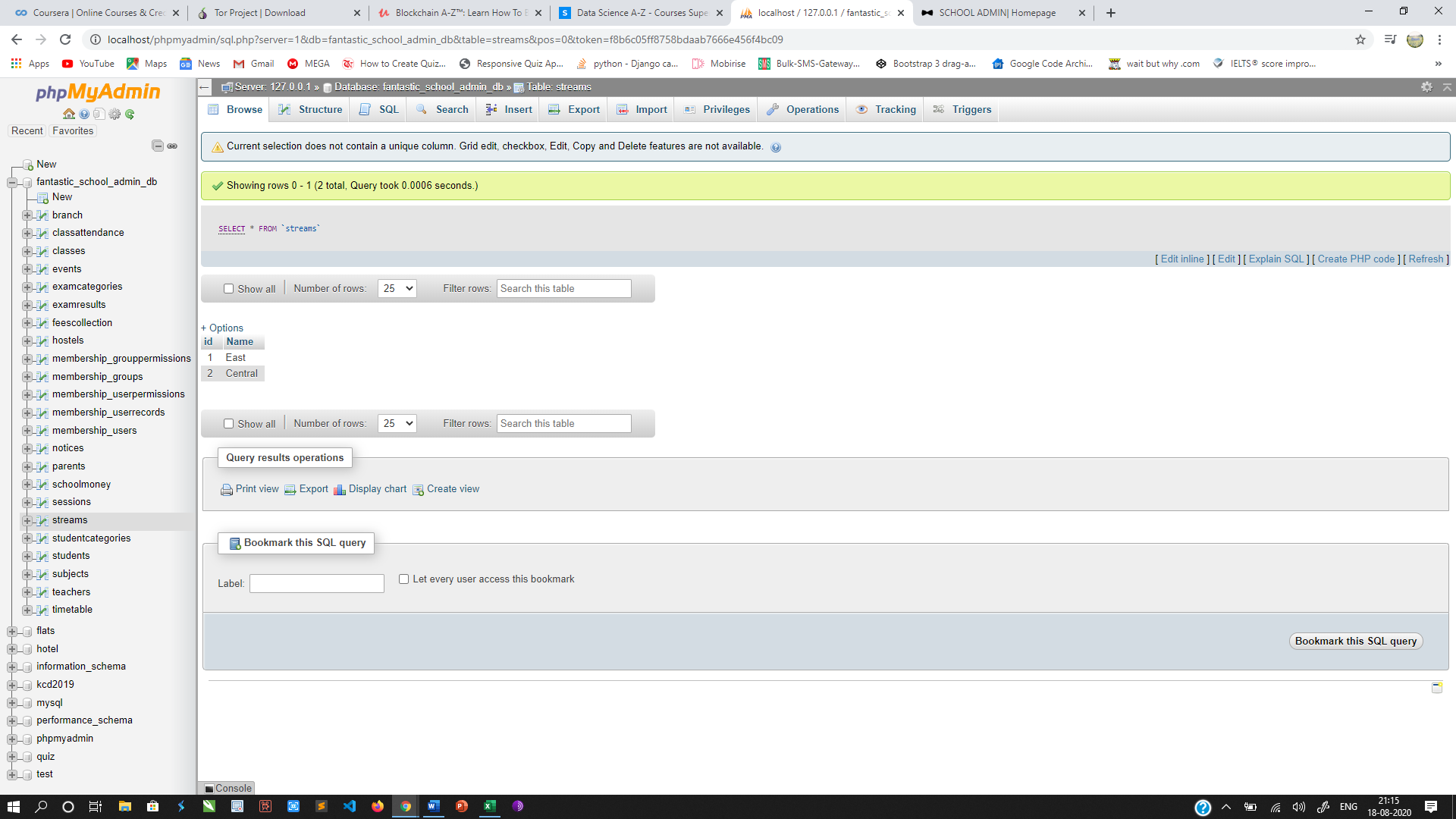


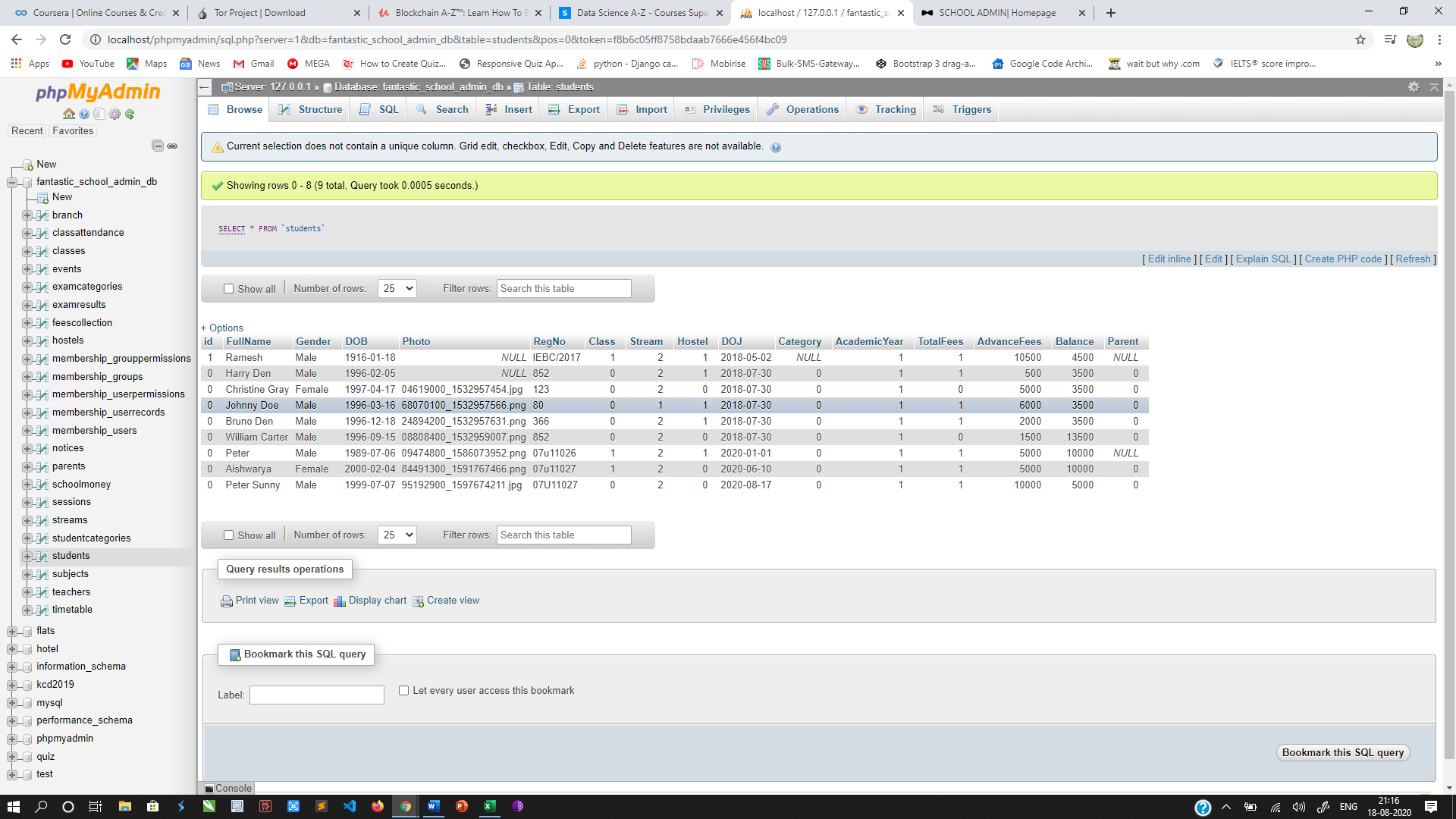


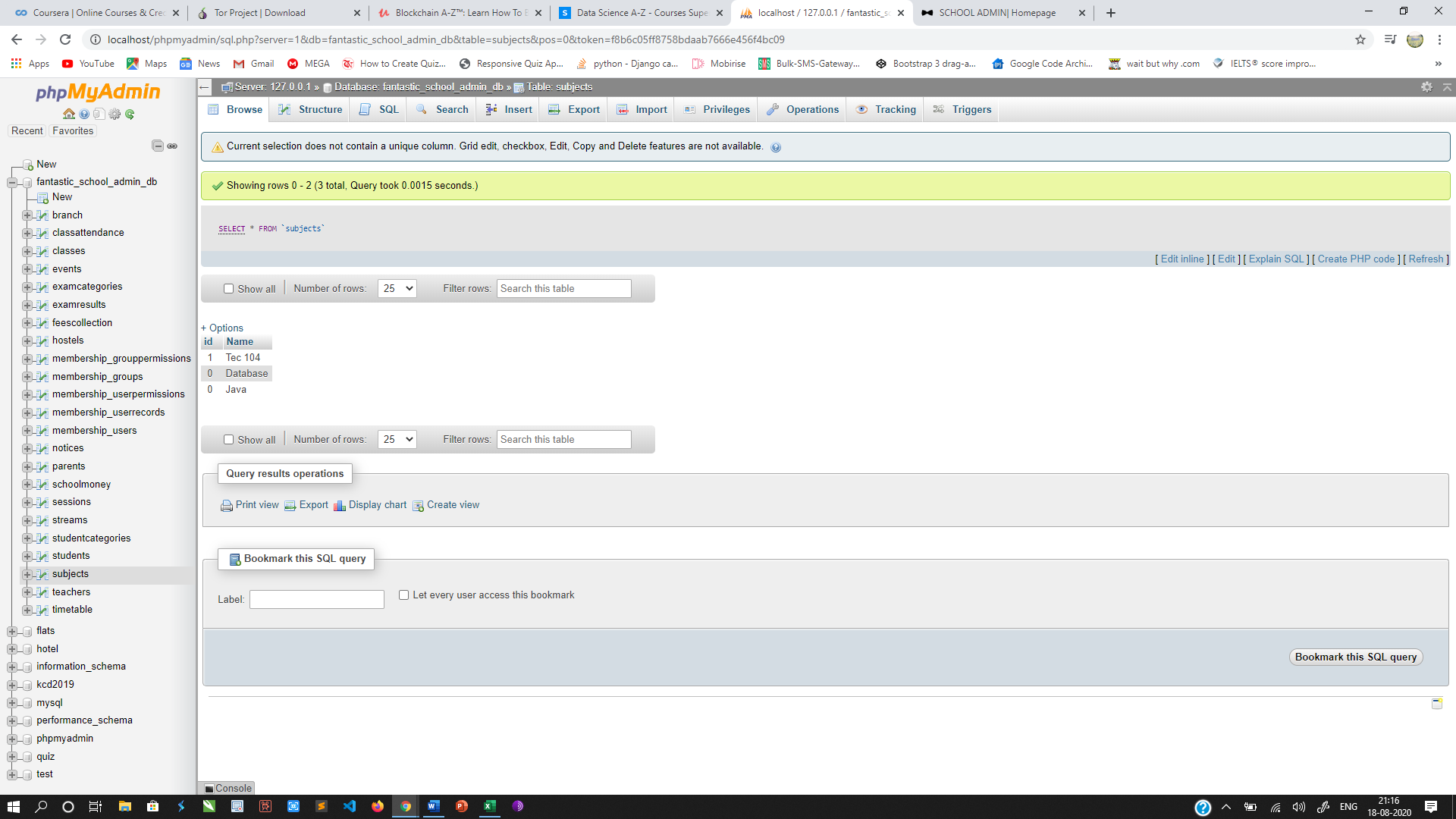


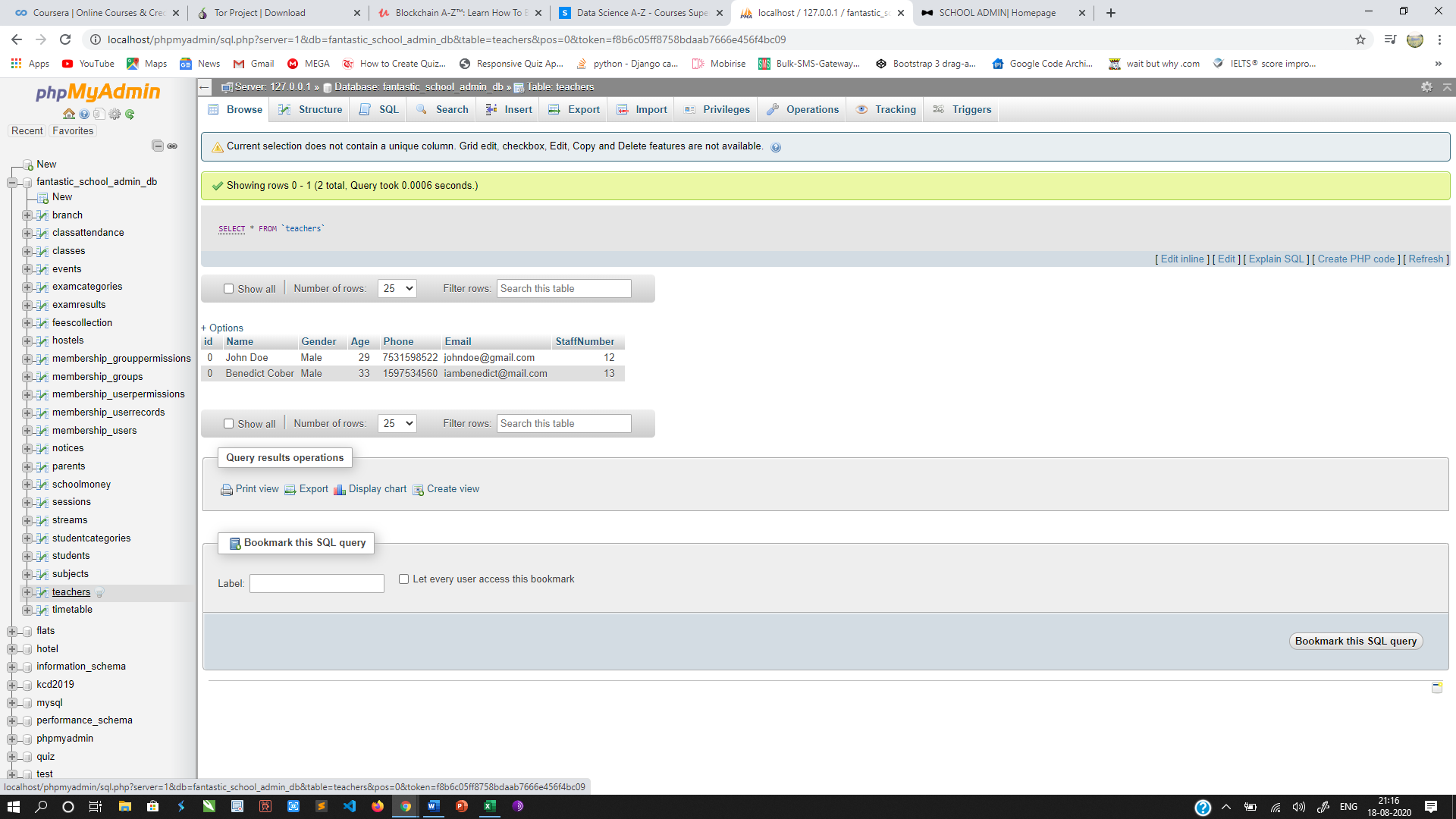






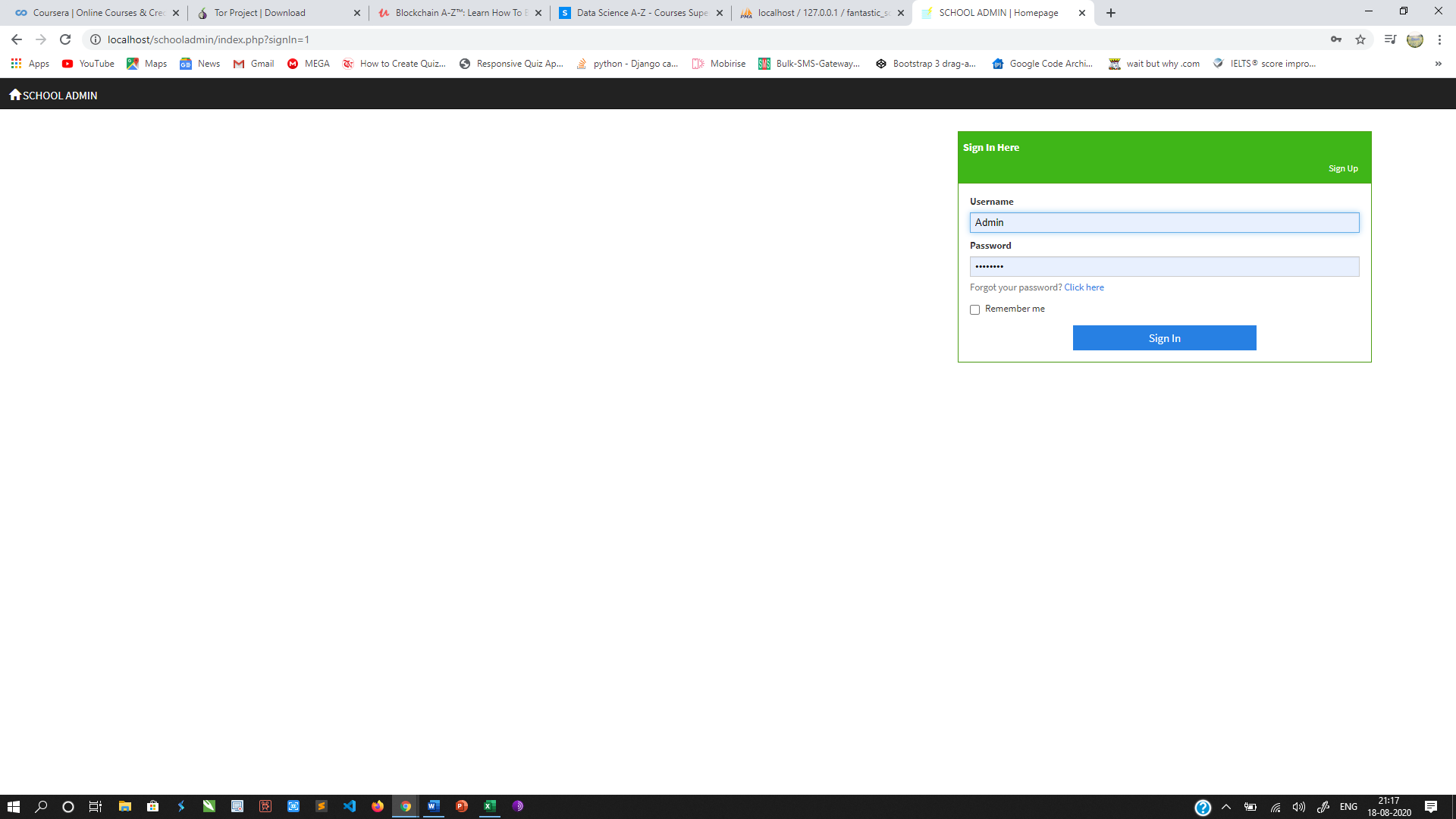




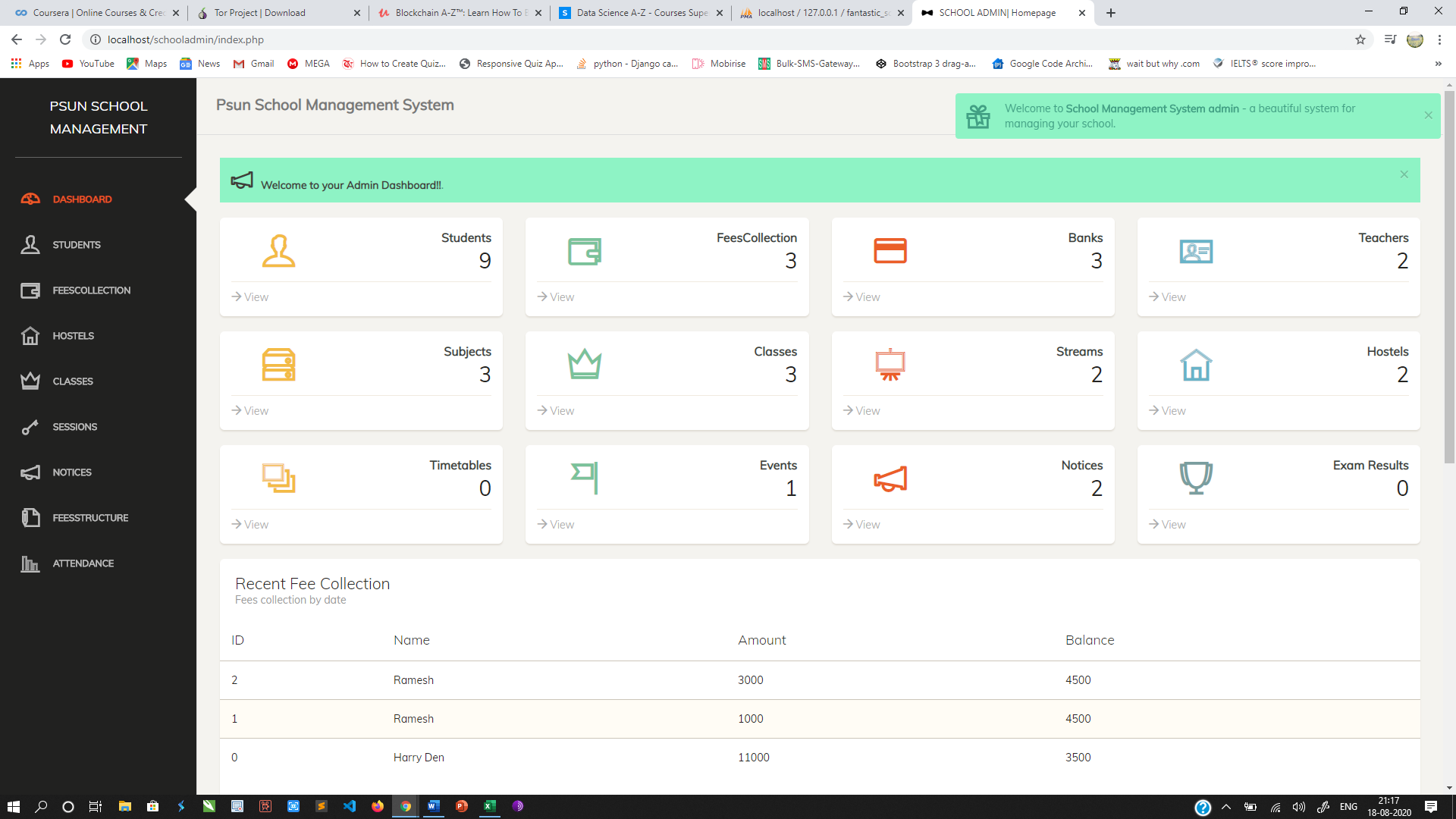


**4.2: Interface Design**

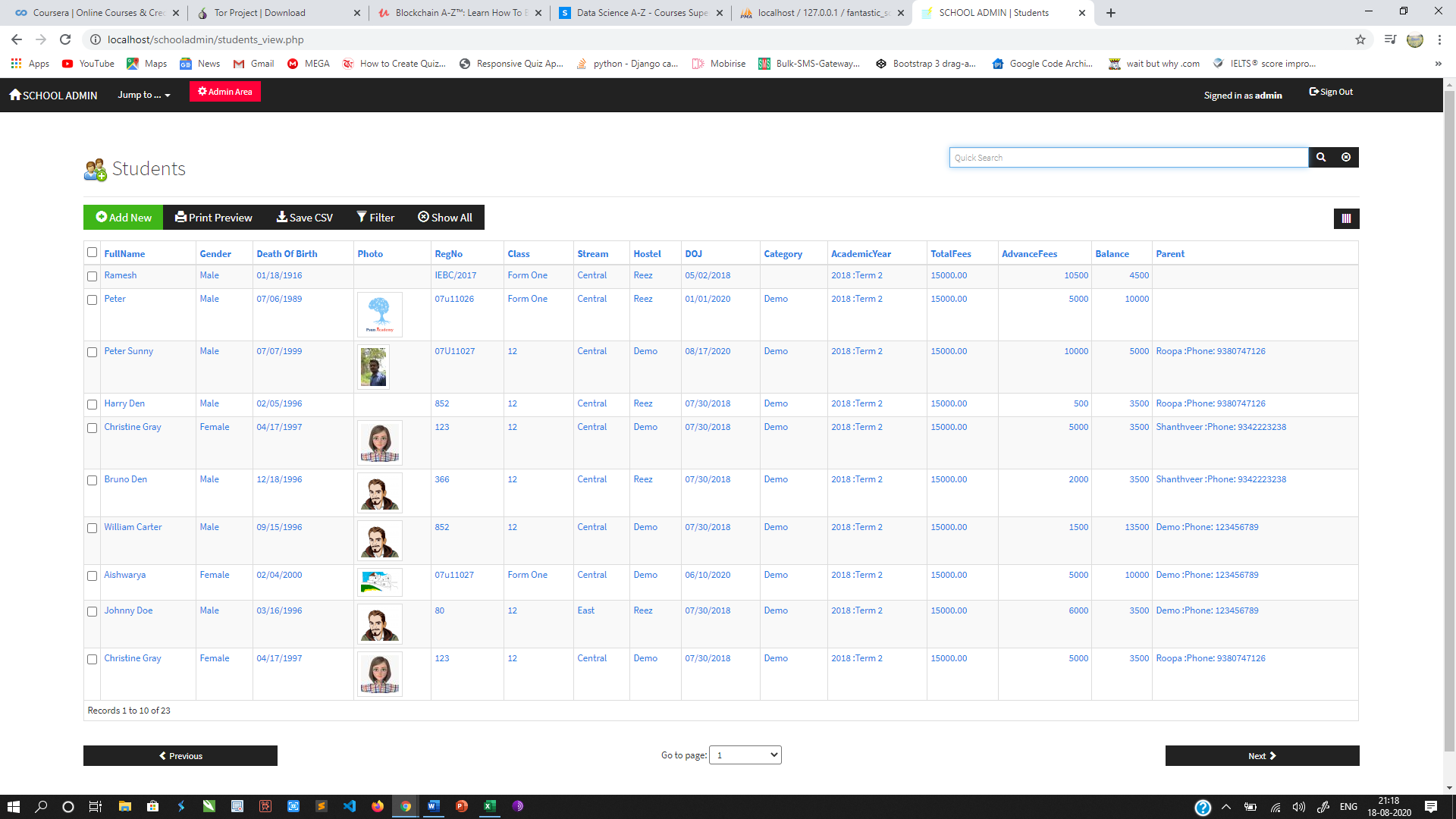
**Login Page**



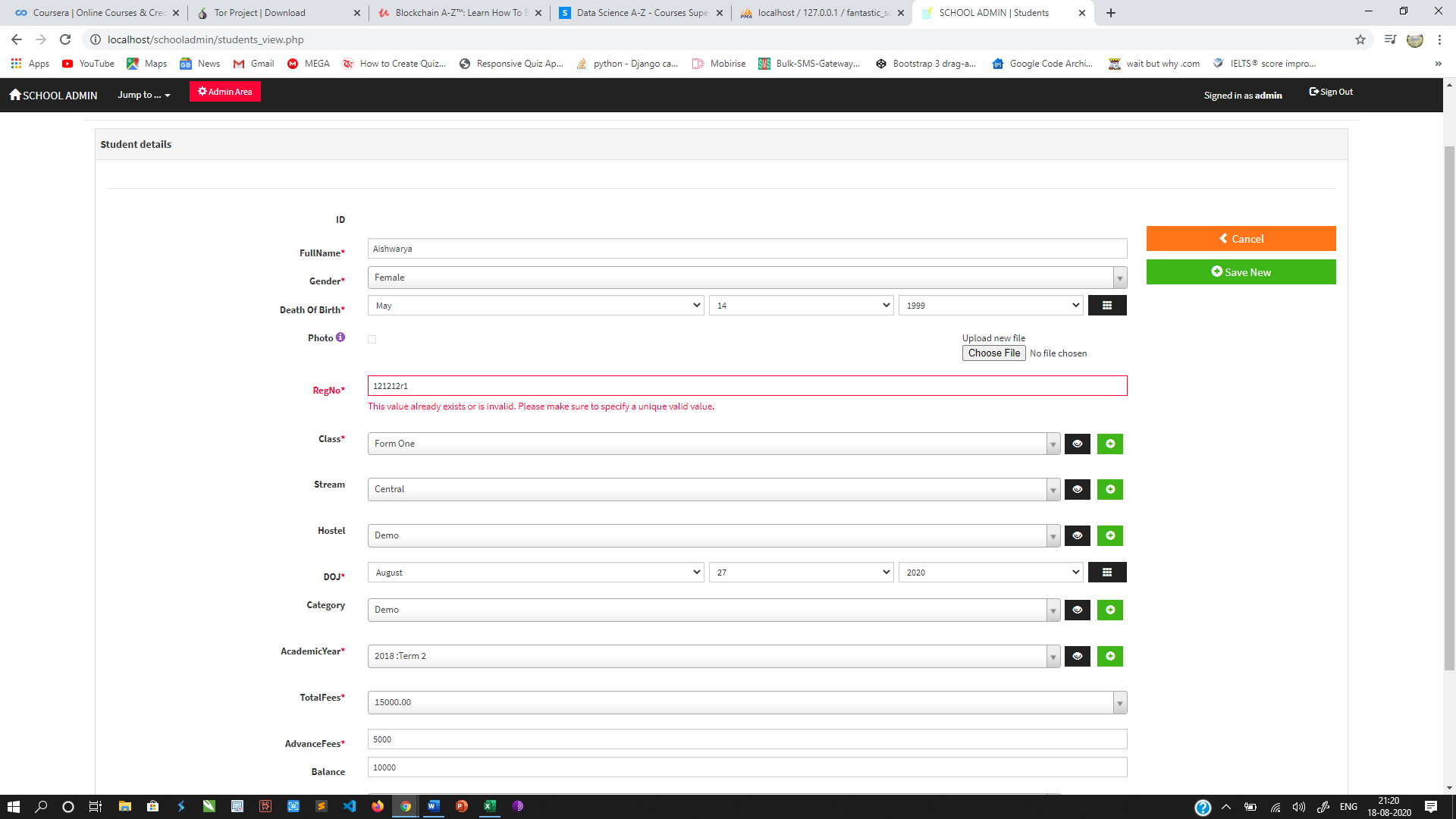
**Dash Board**



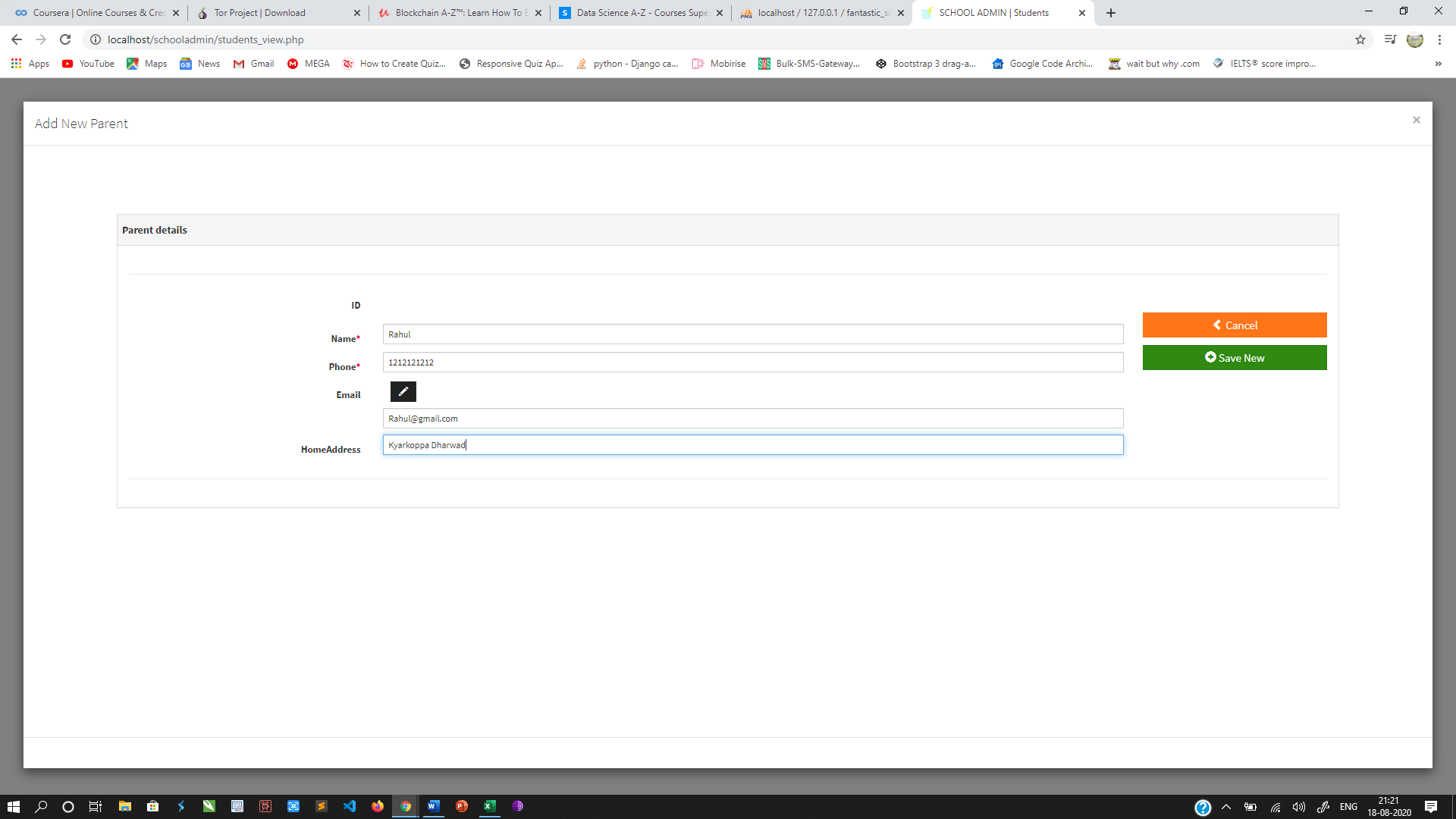
**Students**



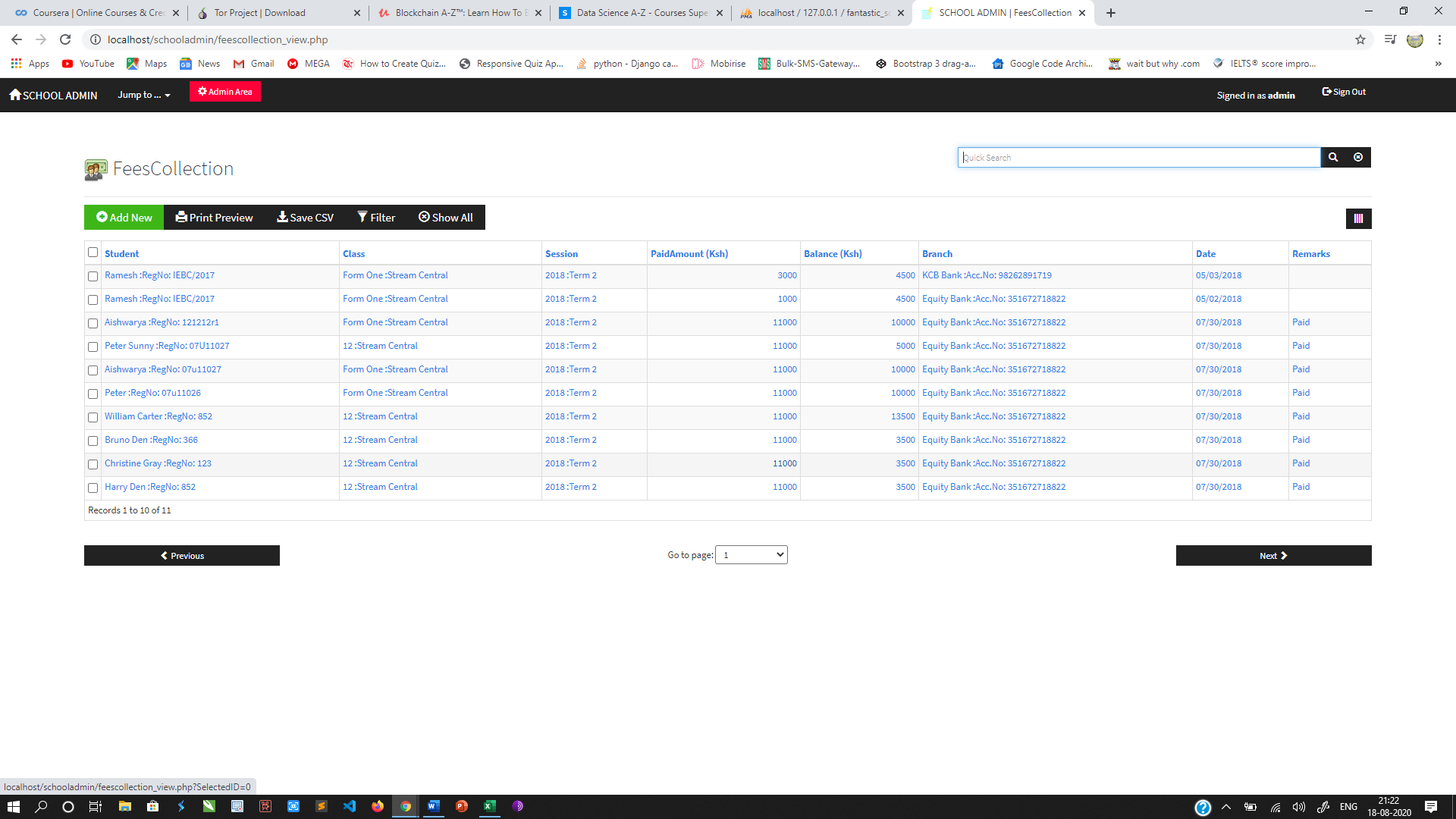
**Add Student Details**



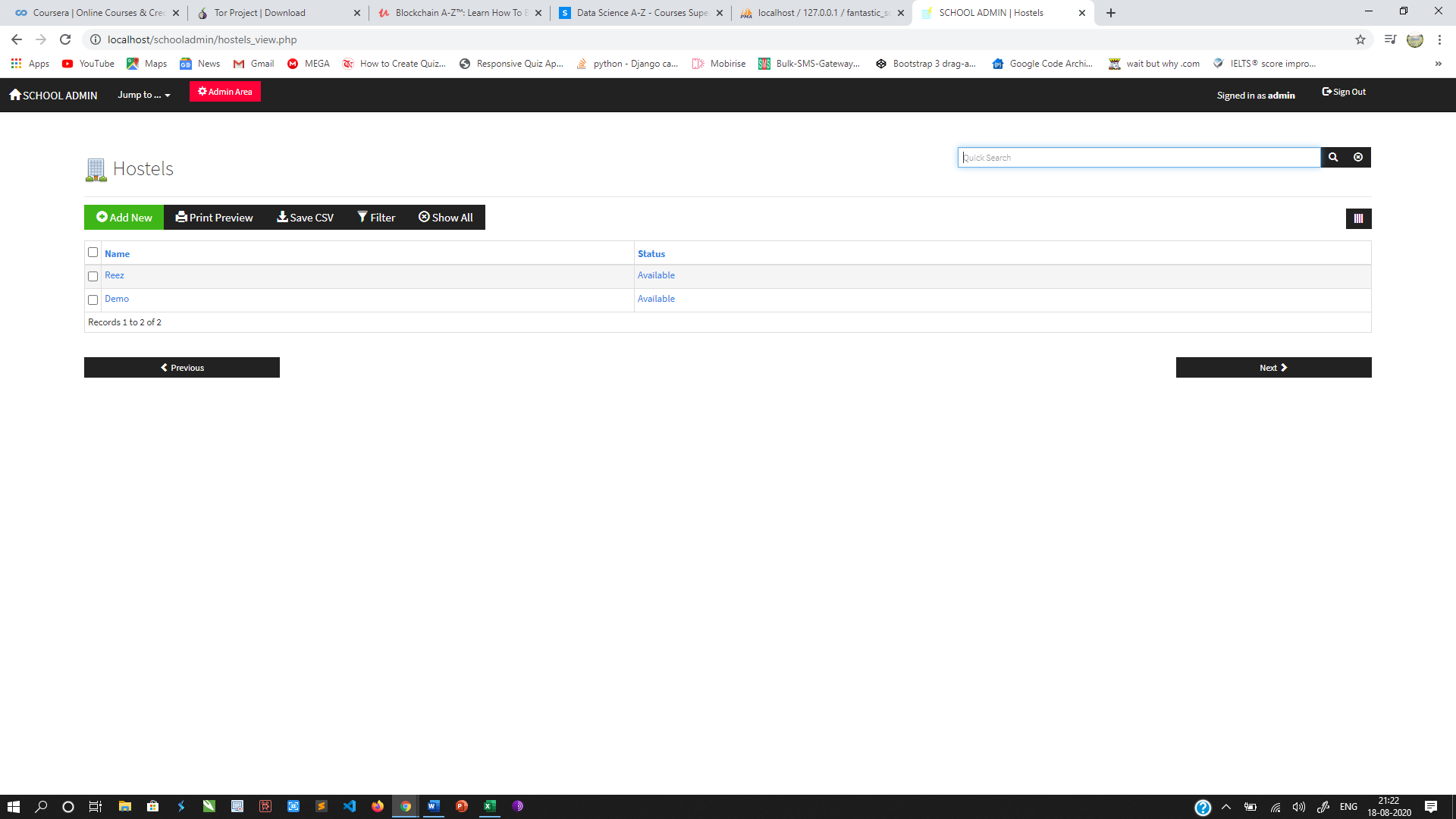
**Add Parent Details of the Student if not available**



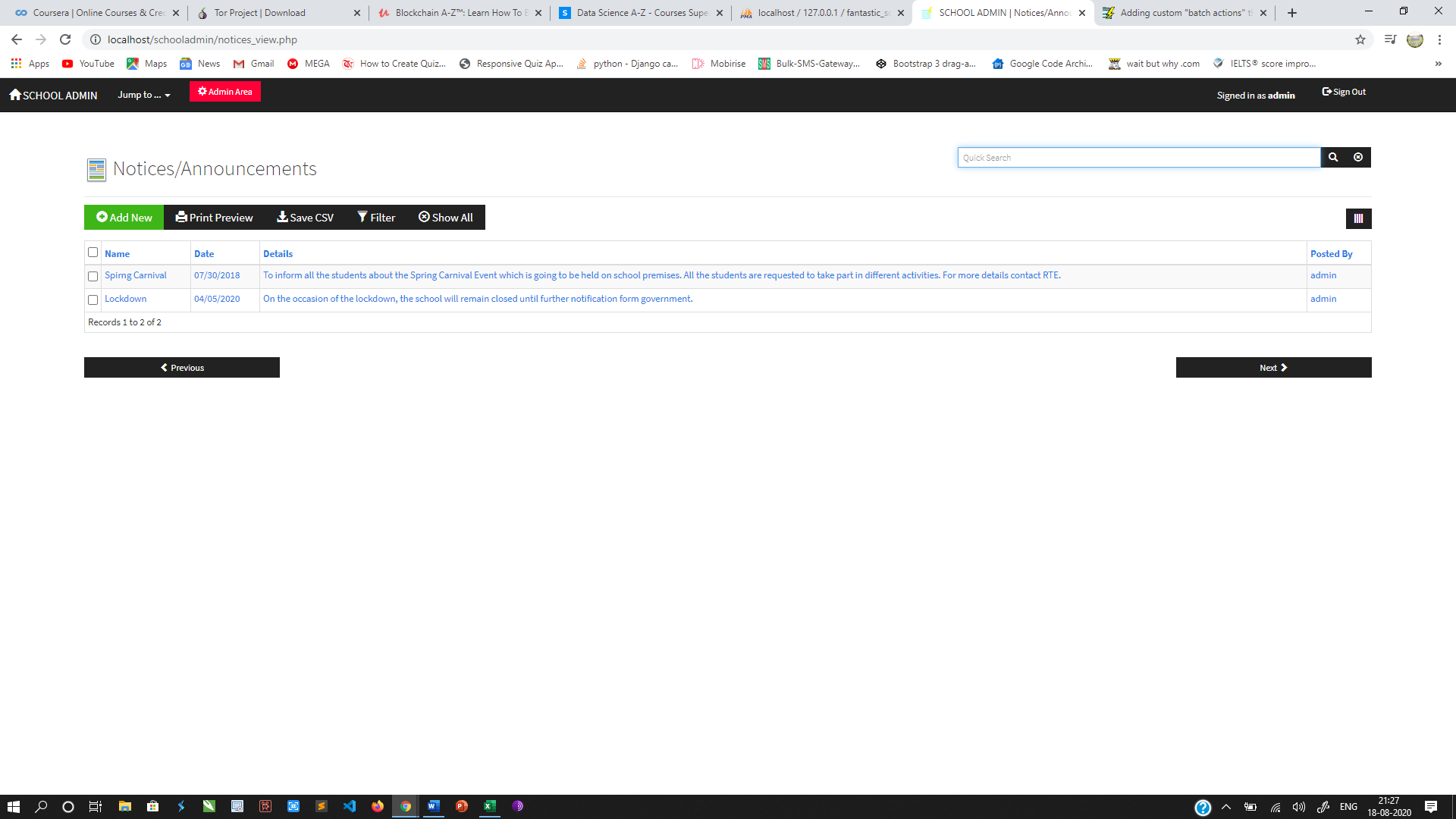
**Fees Collection**



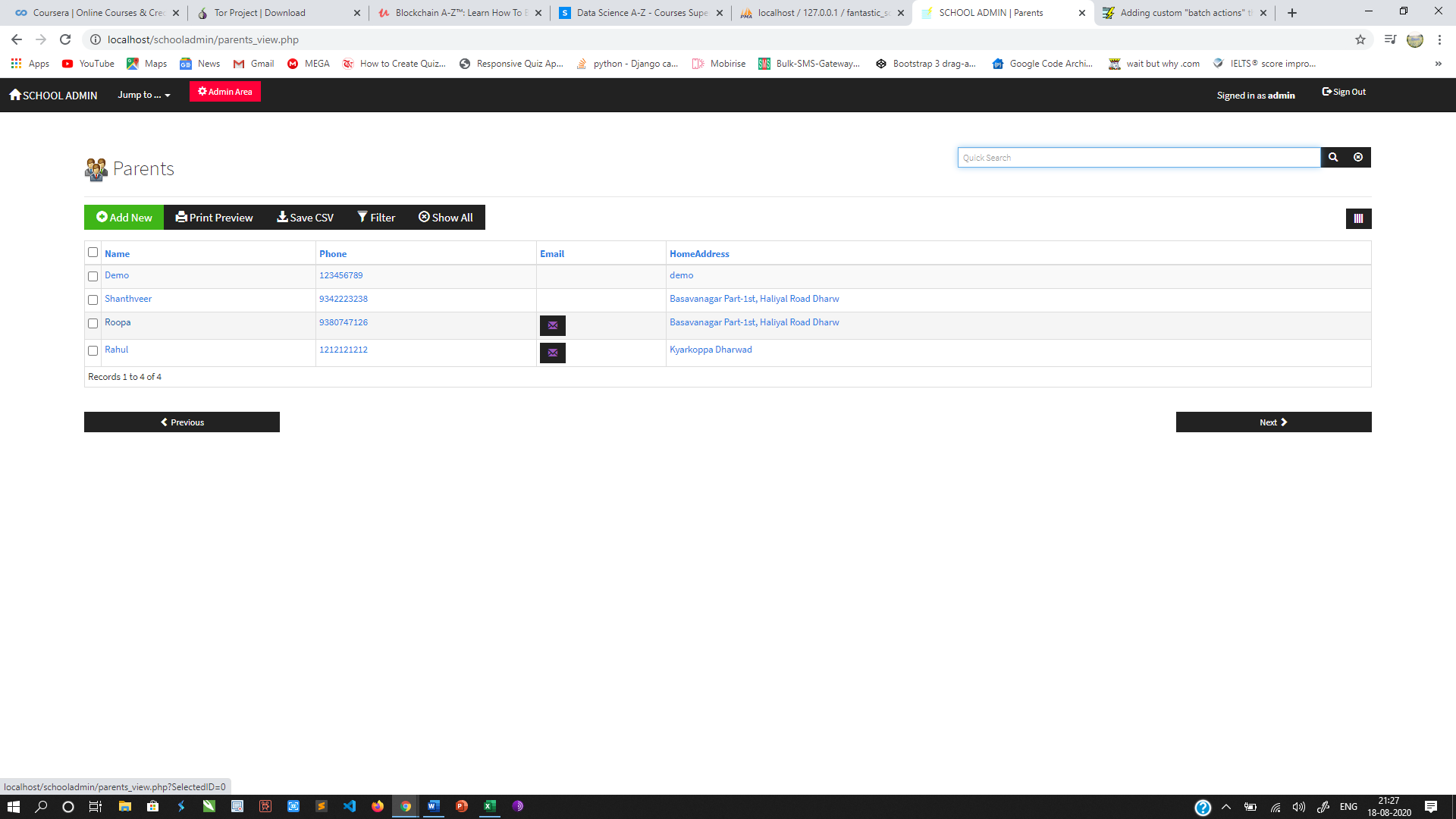
**Hostel Details**



**Notices**



**Parents**



**5 Coding**

**dp.php**

**<?php**

**/\***

**You can implement a new db driver in this file by uncommenting the line below**

**and replacing 'mssql' with the desired driver name, then adding cases for the**

**that driver in each function below ...**

**\*/**

**// define('DATABASE', 'mssql');**

**if(!defined('DATABASE')){**

**if(function\_exists('mysqli\_connect')){**

**define('DATABASE', 'mysqli');**

**}else{**

**define('DATABASE', 'mysql');**

**}**

**}**

**define('mysql\_charset', 'utf8');**

**function db\_link($link = NULL){**

**static $db\_link;**

**if($link) $db\_link = $link;**

**return $db\_link;**

**}**

**function db\_close($link = NULL){**

**db\_link(false); /\* close db link \*/**

**switch(DATABASE){**

**case 'mysql':**

**return mysql\_close($link);**

**case 'mysqli':**

**return mysqli\_close($link);**

**}**

**}**

**function db\_select\_db($dbname, $link = NULL){**

**if(!$link) $link = db\_link();**

**switch(DATABASE){**

**case 'mysql':**

**return mysql\_select\_db($dbname, $link);**

**case 'mysqli':**

**return mysqli\_select\_db($link, $dbname);**

**}**

**}**

**function db\_fetch\_array($res){**

**switch(DATABASE){**

**case 'mysql':**

**return @mysql\_fetch\_array($res);**

**case 'mysqli':**

**return @mysqli\_fetch\_array($res, MYSQLI\_BOTH);**

**}**

**}**

**function db\_fetch\_assoc($res){**

**switch(DATABASE){**

**case 'mysql':**

**return @mysql\_fetch\_assoc($res);**

**case 'mysqli':**

**return @mysqli\_fetch\_assoc($res);**

**}**

**}**

**function db\_fetch\_row($res){**

**switch(DATABASE){**

**case 'mysql':**

**return @mysql\_fetch\_row($res);**

**case 'mysqli':**

**return @mysqli\_fetch\_row($res);**

**}**

**}**

**function db\_num\_fields($res){**

**switch(DATABASE){**

**case 'mysql':**

**return @mysql\_num\_fields($res);**

**case 'mysqli':**

**return @mysqli\_num\_fields($res);**

**}**

**}**

**function db\_num\_rows($res){**

**switch(DATABASE){**

**case 'mysql':**

**return @mysql\_num\_rows($res);**

**case 'mysqli':**

**return @mysqli\_num\_rows($res);**

**}**

**}**

**function db\_affected\_rows($link = NULL){**

**if(!$link) $link = db\_link();**

**switch(DATABASE){**

**case 'mysql':**

**return mysql\_affected\_rows($link);**

**case 'mysqli':**

**return mysqli\_affected\_rows($link);**

**}**

**}**

**function db\_query($query, $link = NULL){**

**if(!$link) $link = db\_link();**

**switch(DATABASE){**

**case 'mysql':**

**return @mysql\_query($query, $link);**

**case 'mysqli':**

**return @mysqli\_query($link, $query);**

**}**

**}**

**function db\_insert\_id($link = NULL){**

**if(!$link) $link = db\_link();**

**switch(DATABASE){**

**case 'mysql':**

**return mysql\_insert\_id($link);**

**case 'mysqli':**

**return mysqli\_insert\_id($link);**

**}**

**}**

**function db\_field\_name($res, $field\_offset){**

**switch(DATABASE){**

**case 'mysql':**

**return @mysql\_field\_name($res, $field\_offset);**

**case 'mysqli':**

**$fo = @mysqli\_fetch\_field\_direct($res, $field\_offset);**

**if($fo === false) return false;**

**return $fo->name;**

**}**

**}**

**function db\_field\_type($res, $field\_offset){**

**switch(DATABASE){**

**case 'mysql':**

**return @mysql\_field\_type($res, $field\_offset);**

**case 'mysqli':**

**$fo = @mysqli\_fetch\_field\_direct($res, $field\_offset);**

**if($fo === false) return false;**

**return $fo->type;**

**}**

**}**

**function db\_escape($str = NULL, $link = NULL){**

**if(!$link) $link = db\_link();**

**switch(DATABASE){**

**case 'mysql':**

**if(function\_exists('mysql\_real\_escape\_string')) return mysql\_real\_escape\_string($str, $link);**

**return mysql\_escape\_string($str);**

**case 'mysqli':**

**return mysqli\_real\_escape\_string($link, $str);**

**}**

**}**

**function db\_connect($host = NULL, $username = NULL, $passwd = NULL, $dbname = NULL, $port = NULL, $socket = NULL){**

**switch(DATABASE){**

**case 'mysql':**

**if($host === NULL) $host = ini\_get("mysql.default\_host");**

**if($username === NULL) $username = ini\_get("mysql.default\_user");**

**if($passwd === NULL) $passwd = ini\_get("mysql.default\_password");**

**$link = mysql\_connect($host, $username, $passwd);**

**if(!$link) return false;**

**db\_link($link); /\* db\_link() can now be used to retrieve the db link from anywhere \*/**

**@mysql\_query("SET NAMES '" . mysql\_charset . "'");**

**if($dbname){ if(!mysql\_select\_db($dbname, $link)) return false; }**

**return $link;**

**case 'mysqli':**

**if($host === NULL) $host = ini\_get("mysqli.default\_host");**

**if($username === NULL) $username = ini\_get("mysqli.default\_user");**

**if($passwd === NULL) $passwd = ini\_get("mysqli.default\_pw");**

**if($dbname === NULL) $dbname = "";**

**if($port === NULL) $port = ini\_get("mysqli.default\_port");**

**if($socket === NULL) $socket = ini\_get("mysqli.default\_socket");**

**$link = mysqli\_connect($host, $username, $passwd, $dbname, $port, $socket);**

**if(!$link) return false;**

**db\_link($link); /\* db\_link() can now be used to retrieve the db link from anywhere \*/**

**mysqli\_set\_charset($link, mysql\_charset);**

**return $link;**

**}**

**}**

**function db\_errno($link = NULL, $mysqli\_connect = false){**

**if(!$link) $link = db\_link();**

**switch(DATABASE){**

**case 'mysql':**

**return mysql\_errno($link);**

**case 'mysqli':**

**if($mysqli\_connect) return mysqli\_connect\_errno();**

**return mysqli\_errno($link);**

**}**

**}**

**function db\_error($link = NULL, $mysqli\_connect = false){**

**if(!$link) $link = db\_link();**

**switch(DATABASE){**

**case 'mysql':**

**return mysql\_error($link);**

**case 'mysqli':**

**if($mysqli\_connect) return mysqli\_connect\_error();**

**return mysqli\_error($link);**

**}**

**}**

**\*\*\*\*\***

Ajax-combo.php (API File Coding)

**<?php**

**// This script and data application were generated by AppGini 5.70**

**// Download AppGini for free from https://bigprof.com/appgini/download/**

**/\***

**ajax-callable script that returns code for either a combo drop-down or an auto-complete**

**drop-down, based on number of items.**

**REQUEST parameters:**

**===============**

**t: table name**

**f: lookup field name**

**id: selected id**

**p: page number (default = 1)**

**s: search term**

**o: 0 (default) for text-only or 1 for full options list**

**text: selected text**

**filterer\_[filterer]: name of filterer field to be used to filter the drop-down contents**

**must be one of the filteres defined for the concerned field**

**\*/**

**$start\_ts = microtime(true);**

**// how many results to return per call, in case of json output**

**$results\_per\_page = 50;**

**$curr\_dir = dirname(\_\_FILE\_\_);**

**include("$curr\_dir/defaultLang.php");**

**include("$curr\_dir/language.php");**

**include("$curr\_dir/lib.php");**

**handle\_maintenance();**

**// drop-downs config**

**$lookups = array(**

**'students' => array(**

**'Class' => array(**

**'parent\_table' => 'classes',**

**'parent\_pk\_field' => 'id',**

**'parent\_caption' => '`classes`.`Name`',**

**'parent\_from' => '`classes` ',**

**'filterers' => array(),**

**'custom\_query' => '',**

**'inherit\_permissions' => true,**

**'list\_type' => 0,**

**'not\_null' => true**

**),**

**'Stream' => array(**

**'parent\_table' => 'streams',**

**'parent\_pk\_field' => 'id',**

**'parent\_caption' => '`streams`.`Name`',**

**'parent\_from' => '`streams` ',**

**'filterers' => array(),**

**'custom\_query' => '',**

**'inherit\_permissions' => true,**

**'list\_type' => 0,**

**'not\_null' => false**

**),**

**'Hostel' => array(**

**'parent\_table' => 'hostels',**

**'parent\_pk\_field' => 'id',**

**'parent\_caption' => '`hostels`.`Name`',**

**'parent\_from' => '`hostels` ',**

**'filterers' => array(),**

**'custom\_query' => 'SELECT `hostels`.`id`, `hostels`.`Name` FROM `hostels` WHERE Status=\'Available\' ORDER BY 2',**

**'inherit\_permissions' => false,**

**'list\_type' => 0,**

**'not\_null' => false**

**),**

**'Category' => array(**

**'parent\_table' => 'studentcategories',**

**'parent\_pk\_field' => 'id',**

**'parent\_caption' => '`studentcategories`.`Name`',**

**'parent\_from' => '`studentcategories` ',**

**'filterers' => array(),**

**'custom\_query' => '',**

**'inherit\_permissions' => true,**

**'list\_type' => 0,**

**'not\_null' => false**

**),**

**'AcademicYear' => array(**

**'parent\_table' => 'sessions',**

**'parent\_pk\_field' => 'id',**

**'parent\_caption' => 'IF(CHAR\_LENGTH(`sessions`.`Year`) || CHAR\_LENGTH(`sessions`.`Term`), CONCAT\_WS(\'\', `sessions`.`Year`, \' :Term \', `sessions`.`Term`), \'\')',**

**'parent\_from' => '`sessions` ',**

**'filterers' => array(),**

**'custom\_query' => 'SELECT `sessions`.`id`, IF(CHAR\_LENGTH(`sessions`.`Year`) || CHAR\_LENGTH(`sessions`.`Term`), CONCAT\_WS(\'\', `sessions`.`Year`, \' :Term \', `sessions`.`Term`), \'\') FROM `sessions` WHERE status=\'active\' ORDER BY 2',**

**'inherit\_permissions' => false,**

**'list\_type' => 0,**

**'not\_null' => true**

**),**

**'TotalFees' => array(**

**'parent\_table' => 'schoolmoney',**

**'parent\_pk\_field' => 'id',**

**'parent\_caption' => '`schoolmoney`.`Total`',**

**'parent\_from' => '`schoolmoney` LEFT JOIN `classes` as classes1 ON `classes1`.`id`=`schoolmoney`.`Class` ',**

**'filterers' => array('Class' => 'Class'),**

**'custom\_query' => '',**

**'inherit\_permissions' => false,**

**'list\_type' => 0,**

**'not\_null' => true**

**),**

**'Parent' => array(**

**'parent\_table' => 'parents',**

**'parent\_pk\_field' => 'id',**

**'parent\_caption' => 'IF(CHAR\_LENGTH(`parents`.`Name`) || CHAR\_LENGTH(`parents`.`Phone`), CONCAT\_WS(\'\', `parents`.`Name`, \' :Phone: \', `parents`.`Phone`), \'\')',**

**'parent\_from' => '`parents` ',**

**'filterers' => array(),**

**'custom\_query' => '',**

**'inherit\_permissions' => true,**

**'list\_type' => 0,**

**'not\_null' => false**

**)**

**),**

**'feescollection' => array(**

**'Student' => array(**

**'parent\_table' => 'students',**

**'parent\_pk\_field' => 'id',**

**'parent\_caption' => 'IF(CHAR\_LENGTH(`students`.`FullName`) || CHAR\_LENGTH(`students`.`RegNo`), CONCAT\_WS(\'\', `students`.`FullName`, \' :RegNo: \', `students`.`RegNo`), \'\')',**

**'parent\_from' => '`students` LEFT JOIN `classes` as classes1 ON `classes1`.`id`=`students`.`Class` LEFT JOIN `streams` as streams1 ON `streams1`.`id`=`students`.`Stream` LEFT JOIN `hostels` as hostels1 ON `hostels1`.`id`=`students`.`Hostel` LEFT JOIN `studentcategories` as studentcategories1 ON `studentcategories1`.`id`=`students`.`Category` LEFT JOIN `sessions` as sessions1 ON `sessions1`.`id`=`students`.`AcademicYear` LEFT JOIN `schoolmoney` as schoolmoney1 ON `schoolmoney1`.`id`=`students`.`TotalFees` LEFT JOIN `parents` as parents1 ON `parents1`.`id`=`students`.`Parent` ',**

**'filterers' => array(),**

**'custom\_query' => '',**

**'inherit\_permissions' => true,**

**'list\_type' => 0,**

**'not\_null' => true**

**),**

**'Class' => array(**

**'parent\_table' => 'students',**

**'parent\_pk\_field' => 'id',**

**'parent\_caption' => 'IF(CHAR\_LENGTH(`students`.`Class`) || CHAR\_LENGTH(`students`.`Stream`), CONCAT\_WS(\'\', IF( CHAR\_LENGTH(`classes1`.`Name`), CONCAT\_WS(\'\', `classes1`.`Name`), \'\'), \' :Stream \', IF( CHAR\_LENGTH(`streams1`.`Name`), CONCAT\_WS(\'\', `streams1`.`Name`), \'\')), \'\')',**

**'parent\_from' => '`students` LEFT JOIN `classes` as classes1 ON `classes1`.`id`=`students`.`Class` LEFT JOIN `streams` as streams1 ON `streams1`.`id`=`students`.`Stream` LEFT JOIN `hostels` as hostels1 ON `hostels1`.`id`=`students`.`Hostel` LEFT JOIN `studentcategories` as studentcategories1 ON `studentcategories1`.`id`=`students`.`Category` LEFT JOIN `sessions` as sessions1 ON `sessions1`.`id`=`students`.`AcademicYear` LEFT JOIN `schoolmoney` as schoolmoney1 ON `schoolmoney1`.`id`=`students`.`TotalFees` LEFT JOIN `parents` as parents1 ON `parents1`.`id`=`students`.`Parent` ',**

**'filterers' => array(),**

**'custom\_query' => '',**

**'inherit\_permissions' => false,**

**'list\_type' => 0,**

**'not\_null' => false**

**),**

**'Session' => array(**

**'parent\_table' => 'sessions',**

**'parent\_pk\_field' => 'id',**

**'parent\_caption' => 'IF(CHAR\_LENGTH(`sessions`.`Year`) || CHAR\_LENGTH(`sessions`.`Term`), CONCAT\_WS(\'\', `sessions`.`Year`, \' :Term \', `sessions`.`Term`), \'\')',**

**'parent\_from' => '`sessions` ',**

**'filterers' => array(),**

**'custom\_query' => 'SELECT `sessions`.`id`, IF(CHAR\_LENGTH(`sessions`.`Year`) || CHAR\_LENGTH(`sessions`.`Term`), CONCAT\_WS(\'\', `sessions`.`Year`, \' :Term \', `sessions`.`Term`), \'\') FROM `sessions` WHERE status=\'active\' ORDER BY 2',**

**'inherit\_permissions' => false,**

**'list\_type' => 0,**

**'not\_null' => true**

**),**

**'Balance' => array(**

**'parent\_table' => 'students',**

**'parent\_pk\_field' => 'id',**

**'parent\_caption' => '`students`.`Balance`',**

**'parent\_from' => '`students` LEFT JOIN `classes` as classes1 ON `classes1`.`id`=`students`.`Class` LEFT JOIN `streams` as streams1 ON `streams1`.`id`=`students`.`Stream` LEFT JOIN `hostels` as hostels1 ON `hostels1`.`id`=`students`.`Hostel` LEFT JOIN `studentcategories` as studentcategories1 ON `studentcategories1`.`id`=`students`.`Category` LEFT JOIN `sessions` as sessions1 ON `sessions1`.`id`=`students`.`AcademicYear` LEFT JOIN `schoolmoney` as schoolmoney1 ON `schoolmoney1`.`id`=`students`.`TotalFees` LEFT JOIN `parents` as parents1 ON `parents1`.`id`=`students`.`Parent` ',**

**'filterers' => array('Session' => 'AcademicYear'),**

**'custom\_query' => '',**

**'inherit\_permissions' => false,**

**'list\_type' => 0,**

**'not\_null' => false**

**),**

**'Branch' => array(**

**'parent\_table' => 'branch',**

**'parent\_pk\_field' => 'id',**

**'parent\_caption' => 'IF(CHAR\_LENGTH(`branch`.`Name`) || CHAR\_LENGTH(`branch`.`AccountNumber`), CONCAT\_WS(\'\', `branch`.`Name`, \' :Acc.No: \', `branch`.`AccountNumber`), \'\')',**

**'parent\_from' => '`branch` ',**

**'filterers' => array(),**

**'custom\_query' => '',**

**'inherit\_permissions' => true,**

**'list\_type' => 0,**

**'not\_null' => true**

**)**

**),**

**'branch' => array(**

**),**

**'teachers' => array(**

**),**

**'subjects' => array(**

**),**

**'classes' => array(**

**),**

**'streams' => array(**

**),**

**'hostels' => array(**

**),**

**'timetable' => array(**

**'Class' => array(**

**'parent\_table' => 'classes',**

**'parent\_pk\_field' => 'id',**

**'parent\_caption' => '`classes`.`Name`',**

**'parent\_from' => '`classes` ',**

**'filterers' => array(),**

**'custom\_query' => '',**

**'inherit\_permissions' => true,**

**'list\_type' => 0,**

**'not\_null' => true**

**),**

**'Stream' => array(**

**'parent\_table' => 'streams',**

**'parent\_pk\_field' => 'id',**

**'parent\_caption' => '`streams`.`Name`',**

**'parent\_from' => '`streams` ',**

**'filterers' => array(),**

**'custom\_query' => '',**

**'inherit\_permissions' => true,**

**'list\_type' => 0,**

**'not\_null' => false**

**)**

**),**

**'events' => array(**

**),**

**'notices' => array(**

**),**

**'examresults' => array(**

**'student' => array(**

**'parent\_table' => 'students',**

**'parent\_pk\_field' => 'id',**

**'parent\_caption' => '`students`.`FullName`',**

**'parent\_from' => '`students` LEFT JOIN `classes` as classes1 ON `classes1`.`id`=`students`.`Class` LEFT JOIN `streams` as streams1 ON `streams1`.`id`=`students`.`Stream` LEFT JOIN `hostels` as hostels1 ON `hostels1`.`id`=`students`.`Hostel` LEFT JOIN `studentcategories` as studentcategories1 ON `studentcategories1`.`id`=`students`.`Category` LEFT JOIN `sessions` as sessions1 ON `sessions1`.`id`=`students`.`AcademicYear` LEFT JOIN `schoolmoney` as schoolmoney1 ON `schoolmoney1`.`id`=`students`.`TotalFees` LEFT JOIN `parents` as parents1 ON `parents1`.`id`=`students`.`Parent` ',**

**'filterers' => array(),**

**'custom\_query' => '',**

**'inherit\_permissions' => true,**

**'list\_type' => 0,**

**'not\_null' => true**

**),**

**'RegNo' => array(**

**'parent\_table' => 'students',**

**'parent\_pk\_field' => 'id',**

**'parent\_caption' => '`students`.`RegNo`',**

**'parent\_from' => '`students` LEFT JOIN `classes` as classes1 ON `classes1`.`id`=`students`.`Class` LEFT JOIN `streams` as streams1 ON `streams1`.`id`=`students`.`Stream` LEFT JOIN `hostels` as hostels1 ON `hostels1`.`id`=`students`.`Hostel` LEFT JOIN `studentcategories` as studentcategories1 ON `studentcategories1`.`id`=`students`.`Category` LEFT JOIN `sessions` as sessions1 ON `sessions1`.`id`=`students`.`AcademicYear` LEFT JOIN `schoolmoney` as schoolmoney1 ON `schoolmoney1`.`id`=`students`.`TotalFees` LEFT JOIN `parents` as parents1 ON `parents1`.`id`=`students`.`Parent` ',**

**'filterers' => array(),**

**'custom\_query' => '',**

**'inherit\_permissions' => false,**

**'list\_type' => 0,**

**'not\_null' => false**

**),**

**'Class' => array(**

**'parent\_table' => 'students',**

**'parent\_pk\_field' => 'id',**

**'parent\_caption' => 'IF( CHAR\_LENGTH(`classes1`.`Name`), CONCAT\_WS(\'\', `classes1`.`Name`), \'\')',**

**'parent\_from' => '`students` LEFT JOIN `classes` as classes1 ON `classes1`.`id`=`students`.`Class` LEFT JOIN `streams` as streams1 ON `streams1`.`id`=`students`.`Stream` LEFT JOIN `hostels` as hostels1 ON `hostels1`.`id`=`students`.`Hostel` LEFT JOIN `studentcategories` as studentcategories1 ON `studentcategories1`.`id`=`students`.`Category` LEFT JOIN `sessions` as sessions1 ON `sessions1`.`id`=`students`.`AcademicYear` LEFT JOIN `schoolmoney` as schoolmoney1 ON `schoolmoney1`.`id`=`students`.`TotalFees` LEFT JOIN `parents` as parents1 ON `parents1`.`id`=`students`.`Parent` ',**

**'filterers' => array(),**

**'custom\_query' => '',**

**'inherit\_permissions' => false,**

**'list\_type' => 0,**

**'not\_null' => false**

**),**

**'Stream' => array(**

**'parent\_table' => 'students',**

**'parent\_pk\_field' => 'id',**

**'parent\_caption' => 'IF( CHAR\_LENGTH(`streams1`.`Name`), CONCAT\_WS(\'\', `streams1`.`Name`), \'\')',**

**'parent\_from' => '`students` LEFT JOIN `classes` as classes1 ON `classes1`.`id`=`students`.`Class` LEFT JOIN `streams` as streams1 ON `streams1`.`id`=`students`.`Stream` LEFT JOIN `hostels` as hostels1 ON `hostels1`.`id`=`students`.`Hostel` LEFT JOIN `studentcategories` as studentcategories1 ON `studentcategories1`.`id`=`students`.`Category` LEFT JOIN `sessions` as sessions1 ON `sessions1`.`id`=`students`.`AcademicYear` LEFT JOIN `schoolmoney` as schoolmoney1 ON `schoolmoney1`.`id`=`students`.`TotalFees` LEFT JOIN `parents` as parents1 ON `parents1`.`id`=`students`.`Parent` ',**

**'filterers' => array(),**

**'custom\_query' => '',**

**'inherit\_permissions' => false,**

**'list\_type' => 0,**

**'not\_null' => false**

**),**

**'Category' => array(**

**'parent\_table' => 'examcategories',**

**'parent\_pk\_field' => 'id',**

**'parent\_caption' => '`examcategories`.`Name`',**

**'parent\_from' => '`examcategories` ',**

**'filterers' => array(),**

**'custom\_query' => '',**

**'inherit\_permissions' => false,**

**'list\_type' => 0,**

**'not\_null' => false**

**),**

**'Subject' => array(**

**'parent\_table' => 'subjects',**

**'parent\_pk\_field' => 'id',**

**'parent\_caption' => '`subjects`.`Name`',**

**'parent\_from' => '`subjects` ',**

**'filterers' => array(),**

**'custom\_query' => '',**

**'inherit\_permissions' => true,**

**'list\_type' => 0,**

**'not\_null' => false**

**),**

**'Term' => array(**

**'parent\_table' => 'sessions',**

**'parent\_pk\_field' => 'id',**

**'parent\_caption' => '`sessions`.`Year`',**

**'parent\_from' => '`sessions` ',**

**'filterers' => array(),**

**'custom\_query' => '',**

**'inherit\_permissions' => true,**

**'list\_type' => 0,**

**'not\_null' => true**

**),**

**'AcademicYear' => array(**

**'parent\_table' => 'students',**

**'parent\_pk\_field' => 'id',**

**'parent\_caption' => 'IF( CHAR\_LENGTH(`sessions1`.`Year`) || CHAR\_LENGTH(`sessions1`.`Term`), CONCAT\_WS(\'\', `sessions1`.`Year`, \' :Term \', `sessions1`.`Term`), \'\')',**

**'parent\_from' => '`students` LEFT JOIN `classes` as classes1 ON `classes1`.`id`=`students`.`Class` LEFT JOIN `streams` as streams1 ON `streams1`.`id`=`students`.`Stream` LEFT JOIN `hostels` as hostels1 ON `hostels1`.`id`=`students`.`Hostel` LEFT JOIN `studentcategories` as studentcategories1 ON `studentcategories1`.`id`=`students`.`Category` LEFT JOIN `sessions` as sessions1 ON `sessions1`.`id`=`students`.`AcademicYear` LEFT JOIN `schoolmoney` as schoolmoney1 ON `schoolmoney1`.`id`=`students`.`TotalFees` LEFT JOIN `parents` as parents1 ON `parents1`.`id`=`students`.`Parent` ',**

**'filterers' => array(),**

**'custom\_query' => '',**

**'inherit\_permissions' => false,**

**'list\_type' => 0,**

**'not\_null' => false**

**)**

**),**

**'parents' => array(**

**),**

**'examcategories' => array(**

**),**

**'sessions' => array(**

**),**

**'studentcategories' => array(**

**),**

**'classattendance' => array(**

**'Subject' => array(**

**'parent\_table' => 'subjects',**

**'parent\_pk\_field' => 'id',**

**'parent\_caption' => '`subjects`.`Name`',**

**'parent\_from' => '`subjects` ',**

**'filterers' => array(),**

**'custom\_query' => '',**

**'inherit\_permissions' => true,**

**'list\_type' => 0,**

**'not\_null' => true**

**),**

**'Student' => array(**

**'parent\_table' => 'students',**

**'parent\_pk\_field' => 'id',**

**'parent\_caption' => '`students`.`FullName`',**

**'parent\_from' => '`students` LEFT JOIN `classes` as classes1 ON `classes1`.`id`=`students`.`Class` LEFT JOIN `streams` as streams1 ON `streams1`.`id`=`students`.`Stream` LEFT JOIN `hostels` as hostels1 ON `hostels1`.`id`=`students`.`Hostel` LEFT JOIN `studentcategories` as studentcategories1 ON `studentcategories1`.`id`=`students`.`Category` LEFT JOIN `sessions` as sessions1 ON `sessions1`.`id`=`students`.`AcademicYear` LEFT JOIN `schoolmoney` as schoolmoney1 ON `schoolmoney1`.`id`=`students`.`TotalFees` LEFT JOIN `parents` as parents1 ON `parents1`.`id`=`students`.`Parent` ',**

**'filterers' => array(),**

**'custom\_query' => '',**

**'inherit\_permissions' => false,**

**'list\_type' => 0,**

**'not\_null' => true**

**),**

**'RegNo' => array(**

**'parent\_table' => 'students',**

**'parent\_pk\_field' => 'id',**

**'parent\_caption' => '`students`.`RegNo`',**

**'parent\_from' => '`students` LEFT JOIN `classes` as classes1 ON `classes1`.`id`=`students`.`Class` LEFT JOIN `streams` as streams1 ON `streams1`.`id`=`students`.`Stream` LEFT JOIN `hostels` as hostels1 ON `hostels1`.`id`=`students`.`Hostel` LEFT JOIN `studentcategories` as studentcategories1 ON `studentcategories1`.`id`=`students`.`Category` LEFT JOIN `sessions` as sessions1 ON `sessions1`.`id`=`students`.`AcademicYear` LEFT JOIN `schoolmoney` as schoolmoney1 ON `schoolmoney1`.`id`=`students`.`TotalFees` LEFT JOIN `parents` as parents1 ON `parents1`.`id`=`students`.`Parent` ',**

**'filterers' => array(),**

**'custom\_query' => '',**

**'inherit\_permissions' => false,**

**'list\_type' => 0,**

**'not\_null' => false**

**),**

**'Class' => array(**

**'parent\_table' => 'students',**

**'parent\_pk\_field' => 'id',**

**'parent\_caption' => 'IF( CHAR\_LENGTH(`classes1`.`Name`), CONCAT\_WS(\'\', `classes1`.`Name`), \'\')',**

**'parent\_from' => '`students` LEFT JOIN `classes` as classes1 ON `classes1`.`id`=`students`.`Class` LEFT JOIN `streams` as streams1 ON `streams1`.`id`=`students`.`Stream` LEFT JOIN `hostels` as hostels1 ON `hostels1`.`id`=`students`.`Hostel` LEFT JOIN `studentcategories` as studentcategories1 ON `studentcategories1`.`id`=`students`.`Category` LEFT JOIN `sessions` as sessions1 ON `sessions1`.`id`=`students`.`AcademicYear` LEFT JOIN `schoolmoney` as schoolmoney1 ON `schoolmoney1`.`id`=`students`.`TotalFees` LEFT JOIN `parents` as parents1 ON `parents1`.`id`=`students`.`Parent` ',**

**'filterers' => array(),**

**'custom\_query' => '',**

**'inherit\_permissions' => false,**

**'list\_type' => 0,**

**'not\_null' => false**

**),**

**'Stream' => array(**

**'parent\_table' => 'students',**

**'parent\_pk\_field' => 'id',**

**'parent\_caption' => 'IF( CHAR\_LENGTH(`streams1`.`Name`), CONCAT\_WS(\'\', `streams1`.`Name`), \'\')',**

**'parent\_from' => '`students` LEFT JOIN `classes` as classes1 ON `classes1`.`id`=`students`.`Class` LEFT JOIN `streams` as streams1 ON `streams1`.`id`=`students`.`Stream` LEFT JOIN `hostels` as hostels1 ON `hostels1`.`id`=`students`.`Hostel` LEFT JOIN `studentcategories` as studentcategories1 ON `studentcategories1`.`id`=`students`.`Category` LEFT JOIN `sessions` as sessions1 ON `sessions1`.`id`=`students`.`AcademicYear` LEFT JOIN `schoolmoney` as schoolmoney1 ON `schoolmoney1`.`id`=`students`.`TotalFees` LEFT JOIN `parents` as parents1 ON `parents1`.`id`=`students`.`Parent` ',**

**'filterers' => array(),**

**'custom\_query' => '',**

**'inherit\_permissions' => false,**

**'list\_type' => 0,**

**'not\_null' => false**

**)**

**),**

**'schoolmoney' => array(**

**'Class' => array(**

**'parent\_table' => 'classes',**

**'parent\_pk\_field' => 'id',**

**'parent\_caption' => '`classes`.`Name`',**

**'parent\_from' => '`classes` ',**

**'filterers' => array(),**

**'custom\_query' => '',**

**'inherit\_permissions' => false,**

**'list\_type' => 0,**

**'not\_null' => true**

**)**

**)**

**);**

**// XSS prevention**

**$xss = new CI\_Input();**

**$xss->charset = datalist\_db\_encoding;**

**// receive and verify user input**

**$table\_name = $\_REQUEST['t'];**

**$field\_name = $\_REQUEST['f'];**

**$search\_id = makeSafe(iconv('UTF-8', datalist\_db\_encoding, $\_REQUEST['id']));**

**$selected\_text = iconv('UTF-8', datalist\_db\_encoding, $\_REQUEST['text']);**

**$returnOptions = ($\_REQUEST['o'] == 1 ? true : false);**

**$page = intval($\_REQUEST['p']);**

**if($page < 1) $page = 1;**

**$skip = $results\_per\_page \* ($page - 1);**

**$search\_term = makeSafe(iconv('UTF-8', datalist\_db\_encoding, $\_REQUEST['s']));**

**if(!isset($lookups[$table\_name][$field\_name])) die('{ "error": "Invalid table or field." }');**

**// can user access the requested table?**

**$perm = getTablePermissions($table\_name);**

**if(!$perm[0] && !$search\_id) die('{ "error": "' . addslashes($Translation['tableAccessDenied']) . '" }');**

**$field = $lookups[$table\_name][$field\_name];**

**$wheres = array();**

**// search term provided?**

**if($search\_term){**

**$wheres[] = "{$field['parent\_caption']} like '%{$search\_term}%'";**

**}**

**// any filterers specified?**

**if(is\_array($field['filterers'])){**

**foreach($field['filterers'] as $filterer => $filterer\_parent){**

**$get = (isset($\_REQUEST["filterer\_{$filterer}"]) ? $\_REQUEST["filterer\_{$filterer}"] : false);**

**if($get){**

**$wheres[] = "`{$field['parent\_table']}`.`$filterer\_parent`='" . makeSafe($get) . "'";**

**}**

**}**

**}**

**// inherit permissions?**

**if($field['inherit\_permissions']){**

**$inherit = permissions\_sql($field['parent\_table']);**

**if($inherit === false && !$search\_id) die($Translation['tableAccessDenied']);**

**if($inherit['where']) $wheres[] = $inherit['where'];**

**if($inherit['from']) $field['parent\_from'] .= ", {$inherit['from']}";**

**}**

**// single value?**

**if($field['list\_type'] != 2 && $search\_id){**

**$wheres[] = "`{$field['parent\_table']}`.`{$field['parent\_pk\_field']}`='{$search\_id}'";**

**}**

**if(count($wheres)){**

**$where = 'WHERE ' . implode(' AND ', $wheres);**

**}**

**// define the combo and return the code**

**$combo = new DataCombo;**

**if($field['custom\_query']){**

**$qm = array(); $custom\_where = ''; $custom\_order\_by = '2';**

**$combo->Query = $field['custom\_query'];**

**if(preg\_match('/ order by (.\*)$/i', $combo->Query, $qm)){**

**$custom\_order\_by = $qm[1];**

**$combo->Query = preg\_replace('/ order by .\*$/i', '', $combo->Query);**

**}**

**if(preg\_match('/ where (.\*)$/i', $combo->Query, $qm)){**

**$custom\_where = $qm[1];**

**$combo->Query = preg\_replace('/ where .\*$/i', '', $combo->Query);**

**}**

**if($where && $custom\_where){**

**$combo->Query .= " {$where} AND ({$custom\_where}) ORDER BY {$custom\_order\_by}";**

**}elseif($custom\_where){**

**$combo->Query .= " WHERE {$custom\_where} ORDER BY {$custom\_order\_by}";**

**}else{**

**$combo->Query .= " {$where} ORDER BY {$custom\_order\_by}";**

**}**

**$query\_match = array();**

**preg\_match('/select (.\*) from (.\*)$/i', $combo->Query, $query\_match);**

**if(isset($query\_match[2])){**

**$count\_query = "SELECT count(1) FROM {$query\_match[2]}";**

**}else{**

**$count\_query = '';**

**}**

**}else{**

**$combo->Query = "SELECT " . ($field['inherit\_permissions'] ? 'DISTINCT ' : '') . "`{$field['parent\_table']}`.`{$field['parent\_pk\_field']}`, {$field['parent\_caption']} FROM {$field['parent\_from']} {$where} ORDER BY 2";**

**$count\_query = "SELECT count(1) FROM {$field['parent\_from']} {$where}";**

**}**

**$combo->table = $table\_name;**

**$combo->parent\_table = $field['parent\_table'];**

**$combo->SelectName = $field\_name;**

**$combo->ListType = $field['list\_type'];**

**if($search\_id){**

**$combo->SelectedData = $search\_id;**

**}elseif($selected\_text){**

**$combo->SelectedData = getValueGivenCaption($combo->Query, $selected\_text);**

**}**

**if($field['list\_type'] == 2){**

**$combo->Render();**

**$combo->HTML = str\_replace('<select ', '<select onchange="' . $field\_name . '\_changed();" ', $combo->HTML);**

**// return response**

**if($returnOptions){**

**?><span id="<?php echo $field\_name; ?>-combo-list"><?php echo $combo->HTML; ?></span><?php**

**}else{**

**?>**

**<span id="<?php echo $field\_name; ?>-match-text"><?php echo $combo->MatchText; ?></span>**

**<input type="hidden" id="<?php echo $field\_name; ?>" value="<?php echo html\_attr($combo->SelectedData); ?>" />**

**<?php**

**}**

**}else{**

**/\* return json \*/**

**header('Content-type: application/json');**

**if(!preg\_match('/ limit .+/i', $combo->Query)){**

**if(!$search\_id) $combo->Query .= " LIMIT {$skip}, {$results\_per\_page}";**

**if($search\_id) $combo->Query .= " LIMIT 1";**

**}**

**$prepared\_data = array();**

**// specific caption provided and list\_type is not radio?**

**if(!$search\_id && $selected\_text){**

**$search\_id = getValueGivenCaption($combo->Query, $selected\_text);**

**if($search\_id) $prepared\_data[] = array('id' => $search\_id, 'text' => $xss->xss\_clean($selected\_text));**

**}else{**

**$res = sql($combo->Query, $eo);**

**while($row = db\_fetch\_row($res)){**

**if(empty($prepared\_data) && $page == 1 && !$search\_id && !$field['not\_null']){**

**$prepared\_data[] = array('id' => empty\_lookup\_value, 'text' => "<{$Translation['none']}>");**

**}**

**$prepared\_data[] = array('id' => iconv(datalist\_db\_encoding, 'UTF-8', $row[0]), 'text' => iconv(datalist\_db\_encoding, 'UTF-8', $xss->xss\_clean($row[1])));**

**}**

**}**

**if(empty($prepared\_data)){ $prepared\_data[] = array('id' => '', 'text' => $Translation['No matches found!']); }**

**echo json\_encode(array(**

**'results' => $prepared\_data,**

**'more' => (@db\_num\_rows($res) >= $results\_per\_page),**

**'elapsed' => round(microtime(true) - $start\_ts, 3)**

**));**

**}**

**6 Conclusion**

The project work titled “School Tool” has been designed using PHP Hypertext Pre-Processor (PHP) where in many userfriendly form controls have been added in order to make it a user interactive application. The system is developed in such a way that the user with common knowledge of computers can handle it easily. The System developed has proved to be user friendly and efficient in achieving basic goals. The system takes care of all the constrains which have specified. The system is found to be really beneficial for the concerned aspects. Application developed is realistic and secure.

 The main idea was to develop an easy to use front end using concept of HTML, CSS, JAVA SCRIPT, and AJAX.

**References**

* Jone, 2015. TOTAL ACCESS FOR RESE. Requirements for Reports on Research Activity For Schools, Departments, Centers & Institutes, 1(2012), p.14.
* Pan, J., 2015. Requirements Analysis for a Student Information System. Student Information System, 1(2003-2004), p.78.
* 10-things-school-management-systems-should-have, n.d. http://www.anseo.net/10-things-school-management-systems-should-have/. [Online].

**7 Bibliography**

brim, 2015. antibullyingsoftware. [Online] Available at:

<https://antibullyingsoftware.com/why-choose-a-web-based-school-management-system/>

simonmlewis, 2015. anseo.net. [Online] Available at:

<http://www.anseo.net/10-things-school-management-systems-should-have/>