**ORGANIZATION MANAGEMENT SYSTEM PROJECT:-**

**Technologies used:-**

**Backend:-** Codeigniter(3.0) with hmvc.

**Frontend:-** Html5, Css 3, javaScript, jQuery (3.3.1),bootstrap 4.1.0,

**Template** :- Admin Lti 3 (link- <https://adminlte.io/themes/dev/AdminLTE/index.html> )

**Modules used** :-

* super\_admin (make admin).
* admin (for dashboard and admin).
* profile (for change username, password and self profile).
* Login.
* account.
* school.
* classes .
* student.
* parents.
* subject.
* category.
* library.
* employee.
* report.
* enquiry.
* settings.
* homework.
* sms.
* email.
* attendance.
* notifications.

**Online url :**- [school.9yug.com](erp.enextit.com)

**Database url :-** [school.9yug.com](erp.enextit.com)**/phpmyadmin**

**Project Description :-** in organization management system it provide management for school, colleges, coaching classes. It can manage students, employee, teacher, fees of school, student invoices, receipt .

**Authorization User-** In this project4 type of users

* Admin (authorization id=1).
* Employee (teacher, professor, accountant) (authorization id=2).
* Parents (authorization id=3).
* Student (authorization id=4).
* Super admin (authorization id=5).

**Super admin**:- Super admin only one , its role create new organization panel, if any customer need this so super admin add organization and details and organization credential is created automatically in db, and if sms service is active so credential send to customer by sms.

**Admin:-**

* Admin can be multiple of same organization. Admin have all the right of edit, delete, create, approve.
* Admin can create, delete, update plan.
* Admin can create employee and give him credential.
* Admin can see attendance report and take attendance.
* Admin can create another admin and give some responsibility to him.
* Admin can change his username & password.

**Employee :-**

* Employee have a role of add student, take attendance, manage library, manage transport.
* Add enquiry if any complain arise.
* Solve ticket and add log in ticket.
* Employee can change his username & password.

**Students:-**

* Student can edit him/her details ,change profile picture etc

**Debugging if error is occure in any page->**

**Example->** <http://school.9yug.net/student/add_student> if you open this page and error is occur then go to ‘your project name’/application/modules/student/controllers/student/add\_student

Here

Student -modules name

student–Controller name

add\_student->method name under controller.

In codeigniter url like- domain name/controller\_name/method name

**Generic model for developer**

in Codeigniter framework location:- project\_name/application/core/MY\_Model

to make code reusable and curd operation easily we make this generic model library,in this we have to pass only parameter with its function name and just extends this model then this will work.

This model is directly extends by CI\_Model (class name is case sensitive in linux server so mbe carefull).

All the table name of whole project displayed here by protected access identifier, so if in future table name will change so we can change here, so no code has been disturbed

**Functions**

* **select($table\_name,$condition,$content\_display,$order)**

Use to fetch data array for table by condition, $content\_display is used to display what data we want for table.

* **insert($table\_name,$params)**

Use to insert data in table.

* **delete($table\_name,$condition)**

Use to delete data by condition in table .

* **update($condition,$table\_name,$params)**

Use to update whole row of table.

* **update\_col($table\_name,$condition,$params)**

Update column by condition and update particular column by $params

* **report($table\_name,$display\_contents,$condition,$start\_date='',$end\_date='',$status='',$coloumn='created\_at')**

give report of table by start date to end date condition wise

* **data\_between\_date($table\_name,$display\_contents,$condition,$interval\_unit,$table\_row='created\_at',$group\_by='',$order\_by='')**

fetch data between date and interval unit like(1 month, 1 year, 7 days).

* **data\_current($table\_name,$display\_contents,$condition,$interval\_unit,$table\_row='created\_at',$start\_date='',$end\_date='')**

fetch data of current month and current year.

* **data\_between\_date\_count($table\_name,$display\_contents,$condition,$interval\_unit='',$start\_date='',$end\_date='')**

fetch number of rows count in table with date andinterval unit like(1 month, 1 year, 7 days).

* **data\_current\_count($table\_name,$display\_contents,$condition,$interval\_unit,$start\_date='',$end\_date='')**

fetch number of rows count in table with date of current year month

* **data\_sum\_between\_date($table\_name,$display\_contents,$condition,$interval\_unit,$coloumn='created\_at',$start\_date='',$end\_date='')**

sum data of table with specific row of between two dates.

* **data\_sum\_current($table\_name,$display\_contents,$condition,$interval\_unit,$coloumn='created\_at',$start\_date='',$end\_date='')**

sum data of table with specific row of between two dates and current year or current month.

For hmvc what is apply on this project-:

* Download hmvc plugin for codeigniter and paste in application/third\_party/.
* Create folder modules in Application/

In application/config/config.php add ($config['modules\_locations'] = array(APPPATH.'modules/' => '../modules/');

Create new module in project (hmvc):-

Go to Application-> modules->Add new folder ->give name(example- admin) lower case

->create 3 folder (controllers,models,views)

* Under controllers create new file with same controller and modules name. controllers file name should start. Example (Admin.php) .
* After create file in controller, extends class with MX\_Controller.
* In this project I create MY\_Controller in application/core which extends with MX\_Controller .we can add authentication process and the process which is common to all the modules.
* Class name always start with capital letter(it is rule).

**NOTE:** If you not follow this structure you phase 404 error because linux server is case sensitive.

**Application/views folder->**

*dashboard :-use to display dashboard.*

*header.php:- all html,head ,css file are here, this view is common to all.*

*Index.php :-* in this file header and footer are included

*$data['\_view'] =’file name’; (example-add\_sales);*

*$this->load->view(‘index',$data);*

**Modules description:-**

***Login module*** :- used to login, logout, forgot password.

Admin Module:- After login if (authorization id=1 and authorization id=2) Admin module dashboard is open.

**Student module :-**In student moduleyou can add student, edit, delete, get student full details .

From student list directly connected to create invoice and receipt.

**subject module :-**.use of subject module is:-

* use to curd operation of subject.
* in which course and batch what subject in list**.**
* Allocate that which teacher/staff/facilty teach particular subject of batch.

**Employee module:-** in this module admin can add employee ,edit ,view ,delete.

**Library module:**- In library module we can add books by detail, add book issue record, view record , book return record add and view .change library setting that what is validity of book issue.

**notification module:**- if we have to send notification or any message to particular student or particular employee or whole student or employee then notification module is used.

**class module :**- In class module we can add academic information like batch and course curd operation.

**homework module :-** In homework module staff or admin can add homework for student to chose batch and courses and he can also select submitted date.

**Enquiry module :-** In enquiry module you can insert fees related enquiry, admission related class related issue , this will work as a ticket system **.**

**transport module :-** transport module is used for bus management add bus driver bus route and in future gps device is add in this**.**

**attendance module :-** in attendance module admin and staff can take attendance of student date wise and batch wise. and also view from date wise.