**ISP 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> )

**For future radius server document:-** <https://freeradius.org/>

**What is radius?**

Remote Authentication Dial-In User Service (RADIUS) is a [client/server](https://searchnetworking.techtarget.com/definition/client-server) [protocol](https://searchnetworking.techtarget.com/definition/protocol) and software that enables remote access servers to communicate with a central server to authenticate dial-in users and authorize their access to the requested system or service. RADIUS allows a company to maintain user profiles in a central [database](https://searchsqlserver.techtarget.com/definition/database) that all remote servers can share. It provides better security, allowing a company to set up a policy that can be applied at a single administered network point. Having a central service also means that it's easier to track usage for billing and for keeping network statistics. Created by Livingston (now owned by Lucent), RADIUS is a de facto industry standard used by a number of network product companies and is a proposed [IETF](https://searchmicroservices.techtarget.com/definition/IETF-Internet-Engineering-Task-Force) standard

**Modules used** :-

* Super\_admin (make admin).
* Admin (for dashboard and admin).
* Profile (for change username, password and self profile).
* Login.
* Account.
* frenchise.
* api\_call.
* recharge.
* Crn ( for add customers)
* Caf.
* Category.
* Payment.
* Staff.
* Report.
* Ticket.
* Log.
* Setting.
* Group.
* Sms.
* Email.
* Task.
* Plan.

**Online url :**- [xportal.ispjet.com](erp.enextit.com)

**Database url :-** [xportal.ispjet.com](erp.enextit.com)**/phpmyadmin**

**Project Description :-** BasicallyThis is Isp Management project is manage customer of isp , create broadband plan ,create customer relation number and caf form, add recharge of customer account, add multiple franchise according to requirement .

This project is like a product it is not only for one user it is multiauth, multiple franchise can use this by privately secure.

This project is used rest api service for curd operation to use this api we use curl library for post and get, In api\_call modules the curl function is create, we have to call api\_call modules

And send url of rest api in post what we have to post.

Modules call by one module to another modules method –

( modules::run(“module name/module controller name/method name”,''rest api url',parameter passed in array if used post method,method name(get,post,delete,put));

Models are not included in this project , because all database operation is handle in rest api side,to make api of any project is helpful if we launch application in many format like mobile,ios,window.so you have to call only url.

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

* Super admin (authorization id=0)
* Admin (authorization id=1)
* Employee((authorization id=2)
* customer(authorization id=3)

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

**Admin:-**

* Admin can be multiple of same frenchise or same company. Admin have all the right of edit, delete, create.
* Admin can create ,delete, update plan.
* Admin can add frenchise under him and five his service to him by share revenue(depeand on %age how much percent share given by frenchise) and fix capacity.
* 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 customer fill caf form and do recharge.
* Add ticket if any complain arise.
* Solve ticket and add log in ticket.
* Employee can change his username & password.

**Customers:-**

* Customer can see his recharge plan, about data uses this will receipt by radius *server (it is remaining because radius server work is not complete) ,*so I used *static data* place of this,
* Customer can view his recharge history.
* Customer can recharge or pay bill by ourself.
* Customer can view his invoice, receipt and ledger details.
* Customer can change his username and password and some personal information.

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

**Example->** <http://xportal.ispjet.com/plan/add_plan> if you open this page and error is occure then go to ‘your project name’/application/modules/plan/controllers/Plan/add\_plan

Here

Plan-modules name

Plan –Controller name

add\_plan->method name under controller.

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

**Generic model for developer**

***It is used in rest api***

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 :-

Method format in controller- snake\_case(example add\_sales())

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.

Plan module -: used to create new plan, edit plan , delete plan in prepaid postpaid and advance rental.

**Api\_call module** :- used to call api by curl library.

**Crn** :- crn(customer relation number) is a unique number of customer when customer add to our company then automatically crn number is generated.

Quick add features is also available in this modules, in quick add features there is multistep form in which crn , caf and what service customer want to use this will generated, plan is optional, at the time of customer creation if customer caf is generated then his credential also automatically generated. And in customer balance table customer balance also generated this will maintain in future transaction of customer if customer recharge and paid bills.

* In one mobile number only one Crn is available .
* In crn list we can directly send we can send sms to number , add ticket to customer, if caf is not added to particular customer then add caf.
* One crn number it is possibility of multiple caf, so no validation is used in create caf.

**Frenchise module: –** one franchise hasauth to addmultiple franchise under him and give him service.

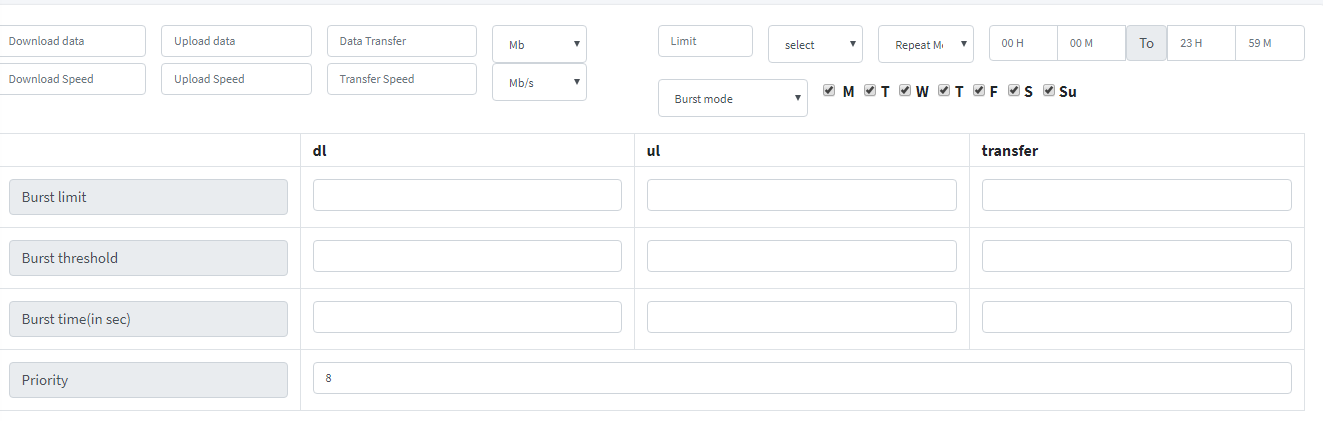
**Nas module :-** Network-attached storage is module is used to curd operation in nas.

**Plan module** :- to make internet plan this module use, at one time you can make multitype plan. (prepaid, postpaid ,advance rental).

You can make unlimited plan to make data transfer 0.

You can make plan like jio like 1 day 1.5 gb data by apply repeat mode=days you can add download and upload speed also.

If you have to make plan that is apply only particular time than add time



You can apply that plan only applicable in particular day.

You can add particular secondary recharge example if (1 gb end so what after 1 gb so click on add more you can add secondary plan)

**Recharge module -:** use to add recharge to customer, but this first add plan otherwise it is not work.

You can see recharge history that which date who do recharge of whose recharge**.**

For postpaid & advance rental users bill paid function are there for prepaid user in recharge panel they can pay.

**Ticket module -:** In Ticket module in generate ticket you can create new ticket of existing customer either new customer, you can see ticket log by ticket no, Send reply to customer.

With the help of this we manage crm.

**Setting module:-** In setting module we can edit our personals details and in billing information you can add your gst number tax percentage which you have to pay,billing cycle means in this day of month automatic bill is created of postpaid and advance rental.

Recharge cycle is cycle in which recharge is automacally done in this date of postpaid and advance rental users.

Franchise terms and condition is displayed in invoice.

In logo upload you can upload your our company logo, it is shown is dashboard and invoice.

**reports module :-** reporting part is important for every project here in reporting part we can check reporting of project like invoice report by date range and invoice status, debit credit report, particular user ledger report by caf id and you can search by date range.

**This project rest api part:-**

**Link-:** xportalapi.ispjet.com

In portal project **application/config/config.**php in api\_ url we set url of rest api in future

If you change domain and server you only change this url.

* The rest api part of this project is made on codeigniter mvc design pattern.
* In this only controller and model, controller takes request and give json response to the server.
* If you wanna make new controller in this so extends the class name with REST\_Controller
* In library REST\_Controller are library file and format.php file that’s why it works as rest api.
* To work with generic model you have to create model of controller and extends with MY\_Model.
* Mostly method of this use camel case method.
* At the last of method \_post or \_get because that is codeigniter rest api use this format

If use post method so \_post at the last of method in controller.