**SYSTEM REPORT**

**PART ONE:**

**Overall architecture of the system**

The system was built with MVC framework, Php 7.4, Apache, Mongo DB, Mongo DB compass for data extraction and visualization.

**File structure:**

---/ mongoDb\_Assignment

---/ app

---/ config

---/ config.php

---/ controllers

---/ Pages.php

---/ Requirements.php

---/ libraries

---/ Controller.php

---/ Core.php

---/ Database.php

---/ model

---/ Index.php

---/ Requirement.php

---/ views

---/ Index.php

---/ requirement

---/ require.php

---/ .htaccess

---/ public

---/ .htaccess

---/ requirement.txt

**Config.php**

This contain database parameters and URL root link to the website

<?php

// this will contain database root app root and file location

// database params

define('DB\_HOST', "mongodb://localhost:27017");

// this is the url root to create dynamic links

define('ASSIGNMENTROOT', 'http://localhost/mongoDb\_Assignment');

**Controller.php**

This will load the load the model and the view the model is connect to the receive data from the controller then query the data from the database, execute the result and return the result to the controller. While the controller grabs the result from the model load the view (the page the user is on) the return data fetched by the user to the view.

<?php

// load the module and the view

class Controller{

    public function model($model){

        // this will require module file

        require\_once '../app/models/' . $model . '.php';

        return new $model();

    }

    // this will load the views and check for the views files

    public function view($view , $data = []){

        if(file\_exists('../app/views/'. $view .'.php')){

            require\_once '../app/views/'. $view .'.php';

        }else{

            die("view  does not exist");

        }

    }

}

**Core.php**

This is the core of the MVC framework which controls the pages , this will check if the page is available then go ahead to load it then it not it will redirect user to the home page.

<?php

// core app class

class Core {

    protected $currentController = 'Pages';

    protected $currentMethod = 'index';

    protected $params = [];

    public function \_\_construct(){

        // print\_r($this->getUrl());

        $url = $this->getUrl();

        // we wil look into our controllers and ucword will capitalize our first letters.

        if(isset($url[0])){

            if(file\_exists('../app/controllers/'. ucwords($url[0]) .'.php')){

                // this will set a new controller

                $this->currentController = ucwords($url[0]);

                unset($url[0]);

            }

        }

        // require the controller

        require\_once '../app/controllers/'  . $this->currentController . '.php';

        // now we extanciate

        $this->currentController = new $this->currentController;

        // this will check for the second parameters

        if(isset($url[1])){

            if(method\_exists($this->currentController, $url[1])){

                $this->currentMethod = $url[1];

                unset($url[1]);

            }

        }

        // this will get the parameters

        $this->params = $url ? array\_values($url) : [];

        // call a callback parameters with arrays of param

        call\_user\_func\_array([$this->currentController, $this->currentMethod], $this->params);

    }

    public function getUrl(){

        if(isset($\_GET['url'])){

            // this will get the url and trim out the slashes

            $url = rtrim($\_GET['url'], '/');

            // this will allow you to filter variable as a string slash number

            $url = filter\_var($url, FILTER\_SANITIZE\_URL);

            // this will explode the url and break it into an array

            $url = explode('/', $url);

            return $url;

        }

    }

}

**Database.php**

This is where the connection occur, this is wrapped around a try and catch block. the database host is passed to the new mongo DB Client. Then some functions are defined to query multiple rows of data **“find ()”** and the other for getting a single row of data **“findOne ()”.** The **“aggregate ()”** function help to count the row and sum the values.

<?php

class Database

{

 private $dbHost = DB\_HOST;

 private $dbHandler;

 public function \_\_construct()

 {

  $options = ["typeMap" => ['root' => 'array', 'document' => 'array']];

  try {

   // database connection

   $this->dbHandler = new MongoDB\Client($this->dbHost, [], $options);

  } catch (\Throwable $e) {

   $e->getMessage();

  }

 }

 /\* create a new method that allows us to create quaries to find all data with parameters

 spacified \*/

 public function find($parameters, $projection)

 {

  // database name

  $dbName =  $this->dbHandler->mongo\_assignment;

  // database collection

  $dbCollection = $dbName->assignment\_collection;

  // return the collection data

  return $dbCollection->find($parameters, $projection);

 }

 /\* create a new method that allows us to create quaries to find one data with parameters

 spacified \*/

 public function findOne($parameters, $projection)

 {

  // database name

  $dbName =  $this->dbHandler->mongo\_assignment;

  // database collection

  $dbCollection = $dbName->assignment\_collection;

  // return the collection data

  return $dbCollection->findOne($parameters, $projection);

 }

 // this will  count  and aggregate data from the mongoDb database

 public function aggregate($parameters)

 {

  // database name

  $dbName =  $this->dbHandler->mongo\_assignment;

  // database collection

  $dbCollection = $dbName->assignment\_collection;

  // return the collection data

  return $dbCollection->aggregate($parameters);

 }

}