1. **WORK PROCESS:**

**Website:**

A website is a collection of web pages on the Internet that you can access using a web browser. It's like an online location where you can find information, shop, connect with others, or enjoy entertainment. Websites can have different purposes and can contain text, images, videos, and interactive elements.

**MVC Framework:**

The *MVC (Model-View-Controller)* framework in *PHP* is a structured approach to web application development. It separates the application's data and business logic (model), user interface (view), and application logic (controller). By using an MVC framework like Laravel, Symfony, or CodeIgniter, developers can organize code, improve maintainability, and build scalable applications more efficiently. These frameworks provide tools for data handling, user interface rendering, and application flow coordination, allowing developers to focus on building robust and modular PHP applications.

MVC framework has three main pillars

**MODEL:** This part connects with the database via *my-sql* code. This is accessed through controller part whenever needed.

**VIEW:** This part shows the view port of the website. This part takes input from user and gives it to the controller through url.

**CONTROLLER:** This part takes input method from url. According to the input it takes data from model and submits it to view or it directly goes to the view site.

show data

if

(posted)

blue line: read data operation

black line: all operation expt read

**FLOW-CHART FOR MVC FRAMEWORK**

index.php

config.php

short form of paths

init.php

autoload class

$rout=new rout()

sets default controller and method

rout.php

reload after user

submit

the form

method name

(from url)

parameters

url function

controller name

(from url)

constructor

default\_view.php

controller

valid ?

method

(parameter)

method

valid ?

yes

yes

no

no

call\_user

\_function

\_array

default method

default\_cont

.php

view method

method

controller.php

data

model

method

model.php

view.php

database

“**index.php”:**

This is the first file where a search engine comes to find the web- page. Here we load

1. “autoload\_class” function from “init.php”
2. Path shortcuts from “config.php”

Then We load rout class.

**“init.php”:**

This file includes “autoload\_class” function. We call this file in “index.php”.

**“config.php”:**

This file includes shortcut paths so that programmer can use these shortcut paths later in programme whenever is needed

Ex:

Define ('DOCUMENTROOT', $\_SERVER['DOCUMENT\_ROOT'] . "/p1");

$\_SERVER['DOCUMENT\_ROOT'] . "/p1" this will be referred as DOCUMENTROOT in the programme.

We call this file in “index.php”.

**“rout.php”:**

This file

1. Defines the default controller and the default method
2. Then, breaks the url in three parts by using “url” method the parts are designed as an array
   1. Url[0] controller
   2. Url[1] method
   3. Url[2] parameter
3. If the controller(Url[0]) file doesn't exist it calls the default controller and default method
4. If the method(Url[1]) doesn't exist inside the controller file it calls the default controller and default method
5. If both the controller and method both exist then it goes to the controller and call the method with the given parameters by using “call\_user\_function\_array” method

**“framework.php”:**

This file defines the structure of “view” function and “model” function & fixes the maximum number of attribute they can have.

1. **View function:** This function leads to the view file causing the show of the webpage
2. **Model function:** This function leads to the model file which interacts with the database.

**“database.php”:**

This file contains the information of the database We have to connect with. It also defines the “Query” function and

“Fetch\_all” function

1. **“Query” function:** This tells us that the sql command executed successfully or not
2. **“Fetch\_all” function:** This fetches all the information from the database

**“controller.php” extends framework:**

The “controller” class is the subclass of framework class. So It can use “view” and “model” function when it is needed. This file works as a connector between viewport, database and user according to the url method request it opens the particular method. That method can have various things like

1. **Go to viewport:** The method can go to a view file to show the web-page.

View Method

1. **POST Forms:** These type of methods have two parts. First it checks that the form is posted or not when we first open the site no form is posted so it goes to the second part and opens the view file to show the webpage which have the form. When The form is submitted by the user the page is reloaded then the first part of the method runs again. It finds that the form is submitted. Then it goes to the model to interact with the database. The model submits the data collected from the user to the database and collects “Query”. Then the second part runs again which leads us to the view file which shows the form and text according the “Query” given by the model.

View Method

If POSTED

no

yes

Model Method

2

database

1

1. **Read Database:** This type of methods also have two parts. The first part goes to the model to read elements from database by “Fetch\_all” function then it comes back to controller goes to the “view” method and shows the data taken from the database in the view file.

View Method

Model Method

2

1

database

1. **Both Read Database and Post Forms:** Firstly It reads data from the database and then it executes the same algorithm as Post Forms.

2

3

yes

Model\_1 Method

If POSTED

1

Model Method

View Method

no

4

database

**“Model.php” extends database:**

The “Model” class is the subclass of database class. The data posted by the user first go to the controller. The controller designs it as an array and posts it to particular model method. By model methods we can do four types of Operation on a Database

1. **ADD:** We can add new data to the database
2. **DELETE:** We can delete existing data from the database
3. **UPDATE:** We can change/update existing data in the database
4. **SHOW:** We can show all the data / requested data in the viewport.

**“View.php”:**

This is the file which shows the viewport to us. All the front-end works are done here.

**How to design structure:**

The whole website file exist inside www folder. It has many folders to organize the website properly

1. **PROJECT:**

Directorate of Ordnance (coordination & Services) [DoO (C&S)] is a prominent organization overseeing Defence Public Sector Undertakings (DPSUs) who are crucial public sector entities dedicated to the production of defence equipment of India. One of the integral components of DoO (C&S) is Intranet, a personal network extensively utilized by headquarters and its units across India. Located in Rajarhaat, their guesthouse provides visitors a comfortable and convenient stay. **As part of my project, I was assigned with creating a website in the Comnet with room booking facilities of the guesthouse online .**

My website follows the “work process” mentioned above. Details of my work is as follows:

1. **Controller file “example.php”:**

This file contains different controller methods

1. **index:** This is the default method. This is a “**go to viewport**” type of method. This goes to view file “**not\_found.php**”
2. **home:** This is a “**go to viewport**” type of method. This leads to “**home.php**” which is the homepage of the website.
3. **contact:** This is a “**go to viewport**” type of method. This leads to “**contact.php**” which is the contact-page of the website.
4. **booking:** This is a “**POST Forms**” type of method. This leads to “**booking.php**” which is the booking-page of the website. When the user submit the form then the method leads to design the submitted data as an array. Then the array is posted to file “**model.php**” to method “**fill\_data**”. When the booking is done the id of that booking is returned to the view.
5. **photo\_galary:** This is a “**go to viewport**” type of method. This leads to “**photo\_galary.php**” which is the gallery page of the website.
6. **approvl\_sts:** This is a “**POST Forms**” type of method. This leads to view file “**approvl\_sts.php**” which is the approval status page of the website. When the user submit the form to see approval status then the method leads to design the submitted data as an array. Then the array is posted to file “**model.php**” to method “**approval**”. The approval status is taken from the database and then posted and printed in “**approvl\_sts.php**”.
7. **review:** This is a “**Both Read Database and Post Forms**” type of method. Firsts this leads to “**model.php**” to method “**show\_reviews.php**”. It takes all the reviews from the database then shows it to the “**review.php**”. When user submits the “review form” then the controller leads to “**model.php**” to method “**reviews**” it submits the review to the database and shows the file “**review.php**”.
8. **admin:** This is a “**Both Read Database and Post Forms**” type of method. First It goes to view file “**admin.php**” which is only accessible by any admin member when the password is put after putting the password in the view file and submitting the form the controller creates the “**msg”** according to the password is right or wrong. Then the message is showed in the view function. Then it brings all the data from database by “**model.php”** method “**admin**” and shows it in the view function “**admin.php**”. When any of the application is approved or not approved by the admin-user then the data given by the user is updated in the database by “**model.php**” method “**admin\_fill”** and the query is shown in “**admin.php**”
9. **Model file “model.php”:**

This file contains different controller methods

1. **fill\_data:**  This takes the array data from the controller method “**booking**”. The data got from controller is included in the database table “**bookingapplication**” by sql command and the query is returned.
2. **approval:** This takes id from the controller method “**approvl\_sts**” and provides the approval status from the database table “**bookingapplication**” to the controller which is to be viewed at view file “**approvl\_sts.php**”
3. **admin:** This takes all the data from the database table “**bookingapplication**” and provides the data to the controller method “**admin**”.
4. **admin\_fill:** This update the data at the database table “**bookingapplication**” when any application is approved or not approved.
5. **show\_reviews:** This file takes all the data from the database table “**review**” and gives it to the controller method “**review**” which is shown in the view file “**review.php”**
6. **reviews:** This file takes input from the “**review .php**” and includes that in the database table “**review**”
7. **View files :**
8. **Header.php:** This file includes all the bootstrap functions which can be used in the programs. This file is included in every view file
9. **Heading.php:** This file includes the heading of the website actually the name of the website. This file is included in every view file except the admin file
10. **Menu.php:** This file is the navigation of the user side to explore the websiteas follows:

|  |  |
| --- | --- |
| **Menu** | **It leads to** |
| **Home** | **home.php** |
| **Booking** | **booking.php** |
| **Approval status** | **approvl\_sts.php** |
| **Review** | **review.php** |
| **Contact** | **contact.php** |
| **Photo Gallery** | **photo\_galary.php** |

1. **home.php:** This view file is called by the controller method “**home**”. This shows the homepage of the site
2. **booking.php:** This is the main part of the website. This part contains a form by submitting which to the controller method “**booking**” user get uniquely generated booking id.
3. **approvl\_sts.php:** This file contains a form where user can put their booking id to check the approval status of their application. The form is posted to the controller method “**booking**”
4. **review.php:** This is a view file which shows the previous reviews and gives user a form to put review. The form get submitted at controller method “**review**”
5. **contact.php:** This view file is called by the controller method “**contact**”. This shows the contact page of the site.
6. **photo\_galary.php:** This view file is called by the controller method “**photo\_galary**”. This shows the contact page of the site.
7. **admin.php:** This view file is only for admin users who can enter the Website by giving a password. Which will be submitted to controller method **“admin**”. Then if it gets the right “**msg**” from the controller method It shows a table with all the details of the application and beside each application there is a place to allot rooms , approve and disapprove booking. The approval buttons ‘gets posted to controller method “**admin**”.

**Creating Web Page**

**For booking of guest house of**

**Directorate of Ordnance (C&S)**

**situated at**

**Rajarhaat, New Town, Kolkata**

By Sreejit Das