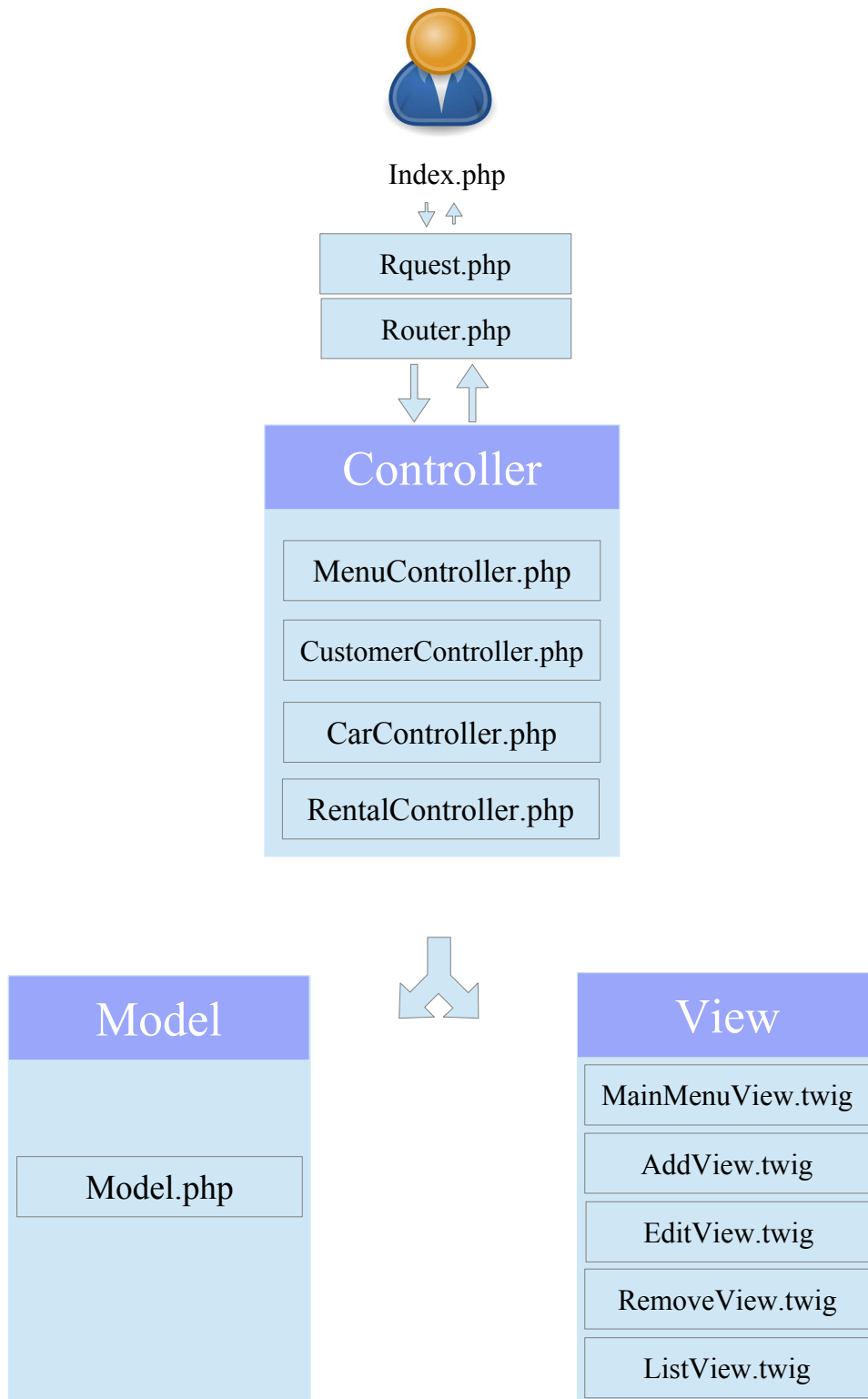
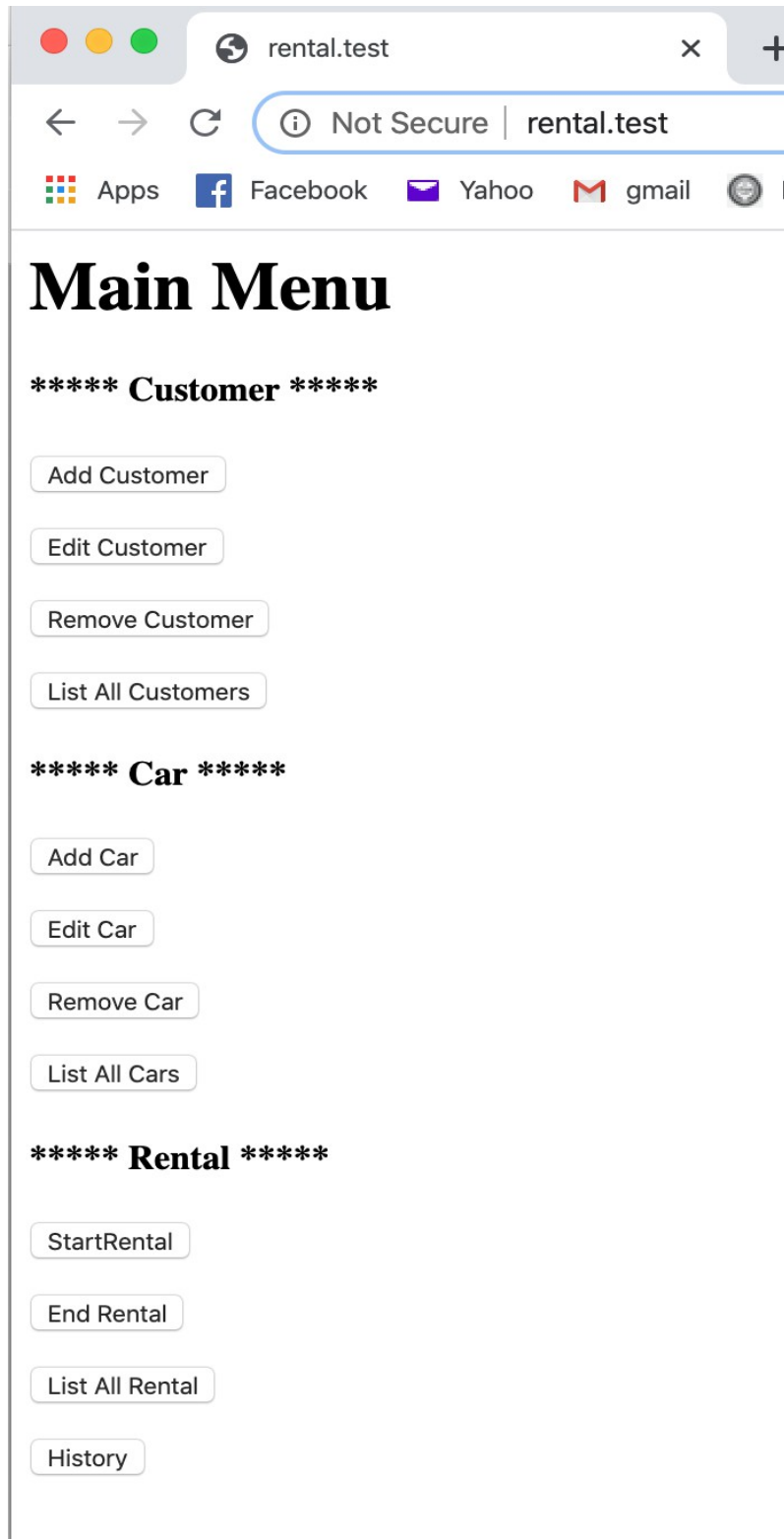


Name: Mustafa Hamraz
Project: Car Rental – u05
GitHub: <https://github.com/mustafa-hamraz/u05-car-rental>





This is how main menu looks like and it consists of three sections:

- Customer
- Car
- Rental

Main Menu

***** Customer *****

Add Customer

Edit Customer

Remove Customer

List All Customers

Add Customer

Person Number:

Name:

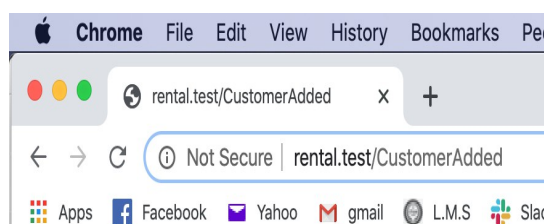
Address:

Postal Address:

Phone Number:

Submit

Main Menu



Customer Added Successfully!

Main Menu

Index.php

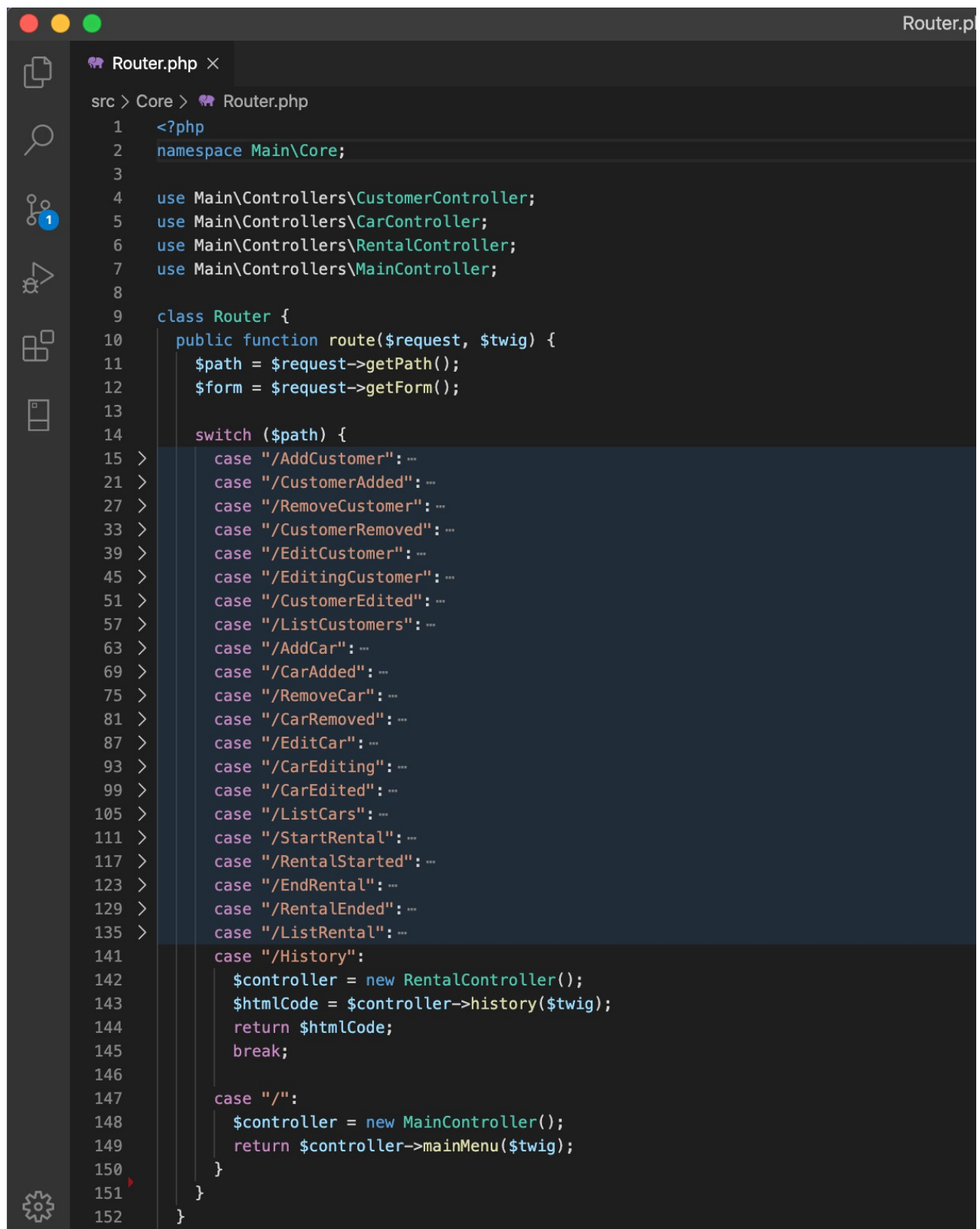
The screenshot shows a code editor interface with a dark theme. The title bar at the top right reads "Index.php — rental". The left sidebar contains an "EXPLORER" panel with a tree view of the project structure. The "OPEN EDITORS" panel shows "Index.php" is open. The "RENTAL" folder is expanded, showing a "src" directory with subfolders "Controllers", "Core", "Models", "Views", and "CarHTML". The "Views" folder is further expanded, showing a list of Twig templates. The main editor area displays the code for "Index.php", which includes PHP tags, namespace imports, and logic for loading Twig templates and rendering a route.

```
1 <?php
2     use Main\Core\Router;
3     use Main\Core\Request;
4
5     require_once __DIR__ . "/vendor/autoload.php";
6     $loader = new Twig_Loader_Filesystem(__DIR__ . "/src/Views");
7     $twig = new Twig_Environment($loader);
8
9     $request = new Request();
10    $router = new Router();
11    $htmlCode = $router->route($request, $twig);
12    echo $htmlCode;
13
14
```

The status bar at the bottom shows "master" and "0 0 0".

Router.php

Router consists of a switch where each case calls upon the right controller base on users request.

A screenshot of a code editor window titled "Router.php". The editor shows the source code for Router.php, which is a PHP file. The code starts with a PHP opening tag and a namespace declaration for "Main\Core". It then uses four classes from the "Main\Controllers" namespace: CustomerController, CarController, RentalController, and MainController. A "Router" class is defined with a public "route" method that takes "\$request" and "\$twig" as arguments. Inside the "route" method, the path is retrieved from the request, and a switch statement is used to handle different paths. The switch statement has cases for various actions like "/AddCustomer", "/CustomerAdded", "/RemoveCustomer", etc., each followed by a comment "...". The case for "/History" creates a new RentalController and calls its "history" method. The case for "/" creates a new MainController and calls its "mainMenu" method. The switch statement ends with a "break;" statement. The code is wrapped in a closing PHP tag and a closing class brace. The editor has a dark theme and a sidebar on the left with icons for file explorer, search, and other IDE features.

```
src > Core > Router.php
1  <?php
2  namespace Main\Core;
3
4  use Main\Controllers\CustomerController;
5  use Main\Controllers\CarController;
6  use Main\Controllers\RentalController;
7  use Main\Controllers>MainController;
8
9  class Router {
10     public function route($request, $twig) {
11         $path = $request->getPath();
12         $form = $request->getForm();
13
14         switch ($path) {
15 >         case "/AddCustomer": ...
21 >         case "/CustomerAdded": ...
27 >         case "/RemoveCustomer": ...
33 >         case "/CustomerRemoved": ...
39 >         case "/EditCustomer": ...
45 >         case "/EditingCustomer": ...
51 >         case "/CustomerEdited": ...
57 >         case "/ListCustomers": ...
63 >         case "/AddCar": ...
69 >         case "/CarAdded": ...
75 >         case "/RemoveCar": ...
81 >         case "/CarRemoved": ...
87 >         case "/EditCar": ...
93 >         case "/CarEditing": ...
99 >         case "/CarEdited": ...
105 >         case "/ListCars": ...
111 >         case "/StartRental": ...
117 >         case "/RentalStarted": ...
123 >         case "/EndRental": ...
129 >         case "/RentalEnded": ...
135 >         case "/ListRental": ...
141         case "/History":
142             $controller = new RentalController();
143             $htmlCode = $controller->history($twig);
144             return $htmlCode;
145             break;
146
147         case "/":
148             $controller = new MainController();
149             return $controller->mainMenu($twig);
150         }
151     }
152 }
```

CarController.php

```
CarController.php — rental

Router.php  CarController.php ×

src > Controllers > CarController.php
1  <?php
2  namespace Main\Controllers;
3
4  use Main\Models\Model;
5
6  class CarController {
7      public function addCar($twig)
8      { ...
11     }
12
13     public function carAdded($twig)
14     { $model = new Model(); ...
33     }
34
35     public function removeCar($twig)
36     { ...
39     }
40
41     public function carRemoved($twig)
42     { ...
56     }
57
58     public function editCar($twig)
59     { ...
62     }
63
64     public function carEditing($twig)
65     { ...
81     }
82
83     public function carEdited($twig)
84     { ...
96     }
97
98     public function listCars($twig)
99     {
100         $model = new Model();
101         $list = $model->db_list_cars();
102
103         $emptyMap = ["list" => $list];
104         return $twig->loadTemplate("CarHTML\ListCarsView.twig")->render($emptyMap);
105     }
106 }
```

Model.php

```
Model.php — rental
Router.php Model.php
src > Models > Model.php
1  <?php
2  namespace Main\Models;
3
4  class Model
5  {
6      /*
7       * Function for connecting to the database.
8       * If you your database username and password is not the same you only need to change here.
9       */private function login()
10     {
11         $servername = "localhost";
12         $dbName = "rental";
13         $username = "homestead";
14         $password = "secret";
15
16         $connection = new \PDO("mysql:host=$servername;dbname=$dbName", $username, $password);
17         if(!$connection) die;
18         return $connection;
19     }
20
21     //Function for adding customer to the database.
22     public function db_add_customer($personNumber, $name, $address, $postalAddress, $phoneNumber)
23     {
24         $conn = $this->login();
25         $sql = 'INSERT INTO customers (person_number, name, address, postal_address, phone_number)
26         VALUES("'.$personNumber.'", "'.$name.'", "'.$address.'", "'.$postalAddress.'", "'.$phoneNumber.'");';
27         $conn->exec($sql);
28     }
29
30     //Function for deleteing customer and it requires a persion_number.
31     public function db_remove_customer($personNumber)
32     { ...
33     }
34
35     public function db_is_customer_renting($personNumber)
36     { ...
37     }
38
39     //Function that checks if a customer exists in the databse and return 0 for false and 1 for true.
40     public function db_exists_customer($personNumber)
41     { ...
42     }
43
44     public function db_return_for_edit_customer($personNumber)
45     { ...
46     }
47
48     //Function that find a customer with id and if it exist it will return all the data about the customer and if not it will return 0
49 }
```

AddCarView.twig

```
Router.php Model.php ● AddCarView.twig ×
src > Views > CarHTML > AddCarView.twig
1  <h1>Add Car</h1>
2
3  <form method="post" action="/CarAdded">
4      Registration Number <input type="text" name="add-reg-number"><p>
5      Make: <input type="text" name="add-make"><p>
6      Color: <input type="text" name="add-color"><p>
7      Year: <input type="text" name="add-year"><p>
8      Price /hour: <input type="text" name="add-price"><p>
9      <input type="submit" value="Submit">
10 </form>
11
12 <form method="post" action="/">
13 |   <input type="submit" value="Main Menu">
14 </form>
15
16
17
```