**ERROR LOG**

1. Creating a network for docker

* docker network create apps-network

1. Failing to create a new database user
2. Dart is null safe . You either must always assign a value or mark it explicitly as nullable using a ?

* String \_prenom = "Something";

or

* String? \_prenom;

1. Bootstrap 4 remove caret from dropdown button

* Simply remove the dropdown-toggle!
* .dropdown-toggle:after { content: none }

1. Wherein clause syntax
2. whereIn('empNo',  $checked\_array)->
3. Checkbox code for inserting only checked rows
4. *//checkbox validation*
5. $checked\_array=$request->input('check');
6. foreach( $request->input('empNo') as $key => $value){
7. if(in\_array ($request->input('empNo')[$key],$checked\_array)){
9. echo  $request->input('empNo')[$key];
10. echo '<br>';
11. echo  $request->input('empName')[$key];
12. echo '<br>';
13. }
14. }
15. Checkbox code for updating only checked rows
16. dd(request());
17. Searching on a specific site

* site:laracasts.com saving multi select

1. How to make an image not cropped on top

* background-position: right 0;

1. Add the following line to composer.Jason

* **"platform-check": false** to avoid the following error
* *Composer detected issues in your platform: Your Composer dependencies require a PHP version ">= 7.3.0"*

1. Perfom the following commands in the terminal

* php artisan config:cache
* composer dump-autoload

1. Model not found error

* They are case sensitive.make sure you write the actual names in the correct case:lower or upper

1. Refused to connect error

* In the .envi file change the value of DB host from 127.0.0.1 to localhost

1. Dockerize a Laravel app

* Create 3 files namely

1. Docker
2. Docker-compose.yaml
3. Default.conf
4. Create custom search form using datatables

For Hiding Default Search Input box of Data tables AS:  
by default **sDom="lftipr";**

Perform these operations on datatables  
'l' - Length changing  
'f' - Filtering input  
't' - The table!  
'i' - Information  
'p' - Pagination  
'r' - pRocessing  
For removing default search box just remove the f character from sDom.

1. Creating a database instance for a database user

* Create a user with no privileges
* Choose a database to which the user will have privileges to
* Edit the privileges for the user

1. How to compress a created image from docker desktop into a .tar file

**command**

* docker save -o ./castel-it-resources\_web.tar castel-it-resources\_web:latest
* where the green part represents the path to save the .tar file
* the yellow part is the name of the tar file
* and the blue part is the actual image to get the tar file from

1. web project resources

* <script src="{{url('js/jquery.min.js')}}"></script>
* <script src="{{url('js/popper.min.js')}}"></script>
* <script src="{{url('js/bootstrap.min.js')}}"></script>
* <link href="{{url('css/bootstrap.min.css')}}" rel="stylesheet" />
* <link rel="stylesheet" href="{{url('fontawesome/css/all.min.css')}}">

1. laravel how to access column with number name of a table?

* $obj->{'22'};

**CREATING A MYSQL SERVICE AND PHPMYADMIN CONTAINER**

*#mysql service*

  mysql:

    image: mysql:8.0

    container\_name: mysql

    volumes:

      - ./data/mysql:/var/lib/mysql

    environment:

      - MYSQL\_ROOT\_PASSWORD=myapp

      - MYSQL\_DATABASE=myapp

      - MYSQL\_USER=myapp

      - MYSQL\_PASSWORD=myapp

    ports:

      - "8001:3306"

*#phpmyadmin service*

  phpmyadmin:

    image: phpmyadmin/phpmyadmin

    container\_name: phpMyAdmin

    links:

      - mysql

    environment:

      PMA\_HOST: mysql

      PMA\_PORT: 3306

      PMA\_ARBITRARY: 1

    restart: always

    ports:

      - 8081:80

1. Remove a docker image

* docker image rm castel-it-resources\_web

1. remove docker container

* docker rm castel-it-resources\_web

1. **“SQLSTATE[HY000] [2054] The server requested authentication method unknown to the client (SQL: select \* from information\_schema.tables where table\_schema = toplayerwins and table\_name = permissions and table\_type = 'BASE TABLE')”**

* ALTER USER 'user'@"localhost" IDENTIFIED WITH mysql\_native\_password BY 'password' ;
* mysql -u root -p

1. .Env file settings
2. DB\_CONNECTION=mysql
3. DB\_HOST=mysqlDB
4. DB\_PORT=3306
5. DB\_DATABASE=castel-resources
6. DB\_USERNAME=root
7. DB\_PASSWORD=P@ssw0rd

* Where DB\_HOST is the name of your mysql container
* DB\_PORT is the port number for your mysql container

1. Checkbox handling
2. @extends('layouts.sectionHead')
3. @section('title') Overtime-automation-system | Verify @endsection
4. @section('content')
5. <form name="names" id="names" method="POST" action="">
7. <div class="container">
8. <div class="table-responsive">
9. <button type="button" class="btn btn-success addmore">Add</button>
10. <button type="button" class="btn btn-danger delete">Remove</button>
11. <br />
12. <table id="demo" class="table table-bordered table-condensed table-striped table-hover">
13. <thead>
14. <tr>
15. <th>
16. <input class="check\_all" type="checkbox" onclick="select\_all()"/>
17. </th>
18. <th>First Name</th>
19. <th>Last Name</th>
20. </tr>
21. </thead>
22. <tbody>
24. <tr>
25. <td><input type="checkbox" name="id[1]" class="case" /></td>
26. <td><input class="form-control" type="text" name="fname[]" id="fname\_1" required></td>
27. <td><input class="form-control" type="text" name="lname[]" id="lname\_1" required></td>
28. </tr>
29. </tbody>
30. </table>
31. <button type="submit" class="btn btn-primary">Submit</button>
32. </form>
33. <br>
34. <br>
35. <br>
36. <div class="submit\_data" style="color:red">click Submit...</div>
38. </div>
39. *<!-- ./tabl*
40. *@endsection*
41. *@section('scripts')*
42. *<script>*
43. */\* Delete Row \*/*
44. *$(".delete").on('click', function() {*
45. *$('.case:checkbox:checked').parents("tr").remove();*
46. *$('.check\_all').prop("checked", false);*
47. *check();*
48. *$('input[name^=id]').on('change', function(e) {*
49. *var thisCheckbox = $(this);*
50. *var thisRow = thisCheckbox.closest('tr');*
51. *if ( thisCheckbox.is(':checked') ) {*
52. *thisRow.addClass('row\_selected');*
53. *} else {*
54. *thisRow.removeClass('row\_selected');*
55. *};*
56. *});*
57. *});*
58. */\* Add Row \*/*
59. *var i=$('table tr').length;*
60. *$(".addmore").on('click',function(){*
61. *count=$('table tr').length;*
62. *var data="<tr><td><input type='checkbox' name='id["+i+"]' class='case'/></td>";*
63. *data += "<td><input class='form-control' id='fname\_"+i+"' name='fname[]' required/></td>";*
64. *data += "<td><input class='form-control' id='lname\_"+i+"' name='lname[]' required/></td></tr>";*
65. *//alert(data);*
67. *$('table').append(data);*
68. */\* Checkbox Select Row Values \*/*
69. *$('input[name^=id]').on('change', function(e) {*
70. *var thisCheckbox = $(this);*
71. *var thisRow = thisCheckbox.closest('tr');*
72. *if ( thisCheckbox.is(':checked') ) {*
73. *thisRow.addClass('row\_selected');*
74. *} else {*
75. *thisRow.removeClass('row\_selected');*
76. *};*
77. *});*
78. *row = i ;*
79. *i++;*
80. *});*
82. */\* Checkbox Select \*/*
83. *function check(){*
84. *obj=$('table tr').find('span');*
85. *$.each( obj, function( key, value ) {*
86. *id=value.id;*
87. *var selected = $('#'+id).html(key+1);*
89. *});*
90. *}*
91. */\* Submit only rows whose checkbox are selected \*/*
92. *$( document ).ready(function() {*
93. *$(function () {*
95. */\* Checkbox Select Row Values \*/*
96. *$('input[name^=id]').on('change', function(e) {*
97. *var thisCheckbox = $(this);*
98. *var thisRow = thisCheckbox.closest('tr');*
99. *if ( thisCheckbox.is(':checked') ) {*
100. *thisRow.addClass('row\_selected');*
101. *} else {*
102. *thisRow.removeClass('row\_selected');*
103. *};*
104. *});*
106. *$("#names").on("submit", function (e) {*
107. *e.preventDefault();*
108. *var toPost = $('.row\_selected input').serialize();*
109. *$('.submit\_data').text(toPost);*
110. *$.ajax({*
111. *type: "POST",*
112. *url: '/verifyconfirm/',*
113. *//     data: $(this).serialize(),*
114. *dataType: 'JSON',*
115. *data: \_token   : '{!! csrf\_field() !!}',*
116. *success: function (response) {*
117. *//alert(data);*
118. *window.location.reload();*
119. *},*
120. *error: function (response) {*
121. *alert(response);*
122. *}*
123. *});*
124. *});*
125. *});*
126. *});*
128. *</script>*
129. *@endsection*
130. Get url value in Laravel
131. request()->route('date')

**Creating a duplicate container instance in portainer**

* Things to change

Volume name

Port number

Container name

**SENDING AN EMAIL FROM A LARAVEL APP**

**Step 1** − We will now send an email from Gmail account and for that you need to configure your Gmail account in Laravel environment file **- .env** file. Enable 2-step verification in your Gmail account and create an application specific password followed by changing the .env parameters as shown below.

.env

MAIL\_DRIVER = smtp

MAIL\_HOST = smtp.gmail.com

MAIL\_PORT = 587

MAIL\_USERNAME = your-gmail-username

MAIL\_PASSWORD = your-application-specific-password

MAIL\_ENCRYPTION = tls

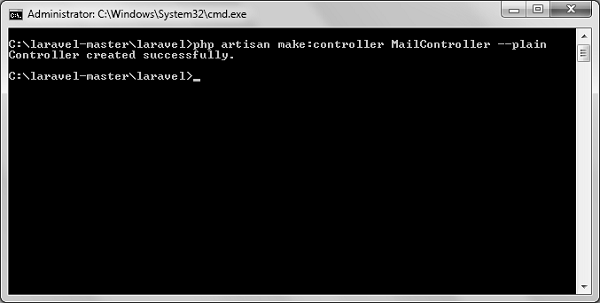
**Step 2** − After changing the **.env** file execute the below two commands to clear the cache and restart the Laravel server.

php artisan config:cache

**Step 3** − Create a controller called **MailController** by executing the following command.

php artisan make:controller MailController –plain

**Step 4** − After successful execution, you will receive the following output −



**Step 5** − Copy the following code in

**app/Http/Controllers/MailController.php** file.

**app/Http/Controllers/MailController.php**

<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;

use Mail;

use App\Http\Requests;

use App\Http\Controllers\Controller;

class MailController extends Controller {

public function basic\_email() {

$data = array('name'=>"Virat Gandhi");

Mail::send(['text'=>'mail'], $data, function($message) {

$message->to('abc@gmail.com', 'Tutorials Point')->subject

('Laravel Basic Testing Mail');

$message->from('xyz@gmail.com','Virat Gandhi');

});

echo "Basic Email Sent. Check your inbox.";

}

public function html\_email() {

$data = array('name'=>"Virat Gandhi");

Mail::send('mail', $data, function($message) {

$message->to('abc@gmail.com', 'Tutorials Point')->subject

('Laravel HTML Testing Mail');

$message->from('xyz@gmail.com','Virat Gandhi');

});

echo "HTML Email Sent. Check your inbox.";

}

public function attachment\_email() {

$data = array('name'=>"Virat Gandhi");

Mail::send('mail', $data, function($message) {

$message->to('abc@gmail.com', 'Tutorials Point')->subject

('Laravel Testing Mail with Attachment');

$message->attach('C:\laravel-master\laravel\public\uploads\image.png');

$message->attach('C:\laravel-master\laravel\public\uploads\test.txt');

$message->from('xyz@gmail.com','Virat Gandhi');

});

echo "Email Sent with attachment. Check your inbox.";

}

}

**Step 6** − Copy the following code in **resources/views/mail.blade.php** file.

**resources/views/mail.blade.php**

<h1>Hi, {{ $name }}</h1>

l<p>Sending Mail from Laravel.</p>

**Step 7** − Add the following lines in **app/Http/routes.php.**

**app/Http/routes.php**

Route::get('sendbasicemail','MailController@basic\_email');

Route::get('sendhtmlemail','MailController@html\_email');

Route::get('sendattachmentemail','MailController@attachment\_email');

**Step 8** − Visit the following URL to test basic email.

<http://localhost:8000/sendbasicemail>

**Step 9** − The output screen will look something like this. Check your inbox to see the basic email output.

Basic Email Sent

**Step 10** − Visit the following URL to test the HTML email.

<http://localhost:8000/sendhtmlemail>

**Step 11** − The output screen will look something like this. Check your inbox to see the html email output.

HTML Email

**Step 12** − Visit the following URL to test the HTML email with attachment.

<http://localhost:8000/sendattachmentemail>

**Step 13** − You can see the following output

Email Sent

**Note** − In the **MailController.php** file the email address in the from method should be the email address from which you can send email address. Generally, it should be the email address configured on your server.

**EXCEL**

1. ALWAYS ALLOW NATIVE DB QUERIES

* In the power query options ,go to the security tab and untick the box that says require user approval to new native db queries

1. How to handle the Formula:firewall error

* In the power query options ,go to the privacy tab and tick always ignore privacy errors

**Flutter**

* Configuring flutter in android studio
* Download the flutter sdk,this will be used to load all your basic structure or code for your new project.
* Extract the flutter sdk to any location and copy its directory and paste it into a new environmental variable called flutter.
* Download the flutter and dart plugin in android studio.
* Important Sites for flutter
* Material.Io/design/color/#tools-for-picking-colors
* Flutter.dev/docs/cookbook

-for Amazing Designing Skills

* Error: Property 'settings' cannot be accessed on 'ModalRoute<Object?>?' because it is potentially null.

**Solution**

data=ModalRoute.of(context)?.settings?.arguments as Map;

NB: The ? is used for null safety.

* [Flutter get User's Country without Manual Permission](https://stackoverflow.com/questions/64173879/flutter-get-users-country-without-manual-permission)

import 'package:http/http.dart' as http;

import 'dart:convert';

Response data = await http.get('http://ip-api.com/json');

Map data = jsonDecode(data.body);

String country = data['country'];

* How to upload /sync you project to GitHub
* Open GitHub desktop app
* Click on add Repository
* Click on create new repository
* Enter the name and description **NB** the name should be the same as the folder name in your projects folder
* Locate the folder containing your android project name
* Click on initialize repository
* Click create repository
* Click on publish repository

CANTEEN MANAGEMENT SYSTEM

SERVER CONNECTION

[How to solve "Error: MySQL shutdown unexpectedly"?](https://stackoverflow.com/questions/18022809/how-to-solve-error-mysql-shutdown-unexpectedly)

**IMPORTANT:** do **NOT** delete ibdata1 file. You could destroy all your databases.

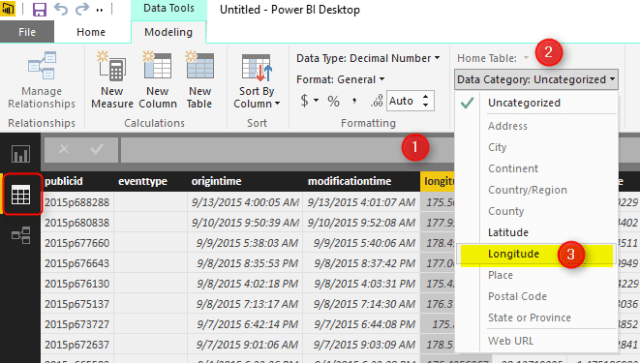
Instead, first try using the **MySQL backup folder** which is included with XAMPP. So do next steps:

1. Rename folder mysql/data to mysql/data\_old
2. Make a copy of mysql/backup folder and name it as mysql/data
3. Copy all your database folders from mysql/data\_old into mysql/data (except mysql, performance\_schema, and phpmyadmin folders)
4. Copy mysql/data\_old/ibdata1 file into mysql/data folder
5. Start MySQL from XAMPP control panel

POWER BI

* Using latitudes and longtudes

As you see Power BI considered these fields as numeric fields and summarized them in the view (you can see sigma icon besides them in screenshot above). This can be easily changed in the data tab. Select the Column and then change the Data Category under the Modeling menu to respective field (latitude or longitude).

[](https://i2.wp.com/radacad.com/wp-content/uploads/2015/09/314.png)

Now go back to Report tab. This time you will see these two fields with map icons besides them.

# **the on-premises data gateway installation has failed**

Please take a try to open secpol.msc, then find "***Create global objects***" policy as mentioned within above thread, and then create your own account within it, check if the issue is solved.