



Introduction

In this tutorial exercise, we will learn using the video tutorial “Laravel CRUD in 50 minutes for Beginners from Scratch” by Dectamin (<https://www.youtube.com/watch?v=LA9QsgJ0bw>).

After the 30-minute tutorial in the previous experiment with CodeIgniter, you should have a basic understanding of the web flow and lifecycle with PHP. Now, you will try another framework to compare its design and functionality.

Exercises

1. Watch the video tutorial linked above.
2. Build an email client with Create, Retrieve, and Delete functions for emails.
3. The input fields in the user interface should include: To, CC, BCC, Subject, Message, and a "Send" button. Other appropriate buttons or links should be provided to navigate between pages, such as "Back" and "List Email".
4. The email client should be controlled by an "EmailController".
5. The email message should be represented by an "Email" model.

Submission

After completion, provide the following evidence for participation marks.

Screenshots of:

1. The “List Email” interface.
2. The “Create Email” interface.
3. The “Read Email” message interface.
4. The source code in EmailController.
5. The source code in the “Email” model.

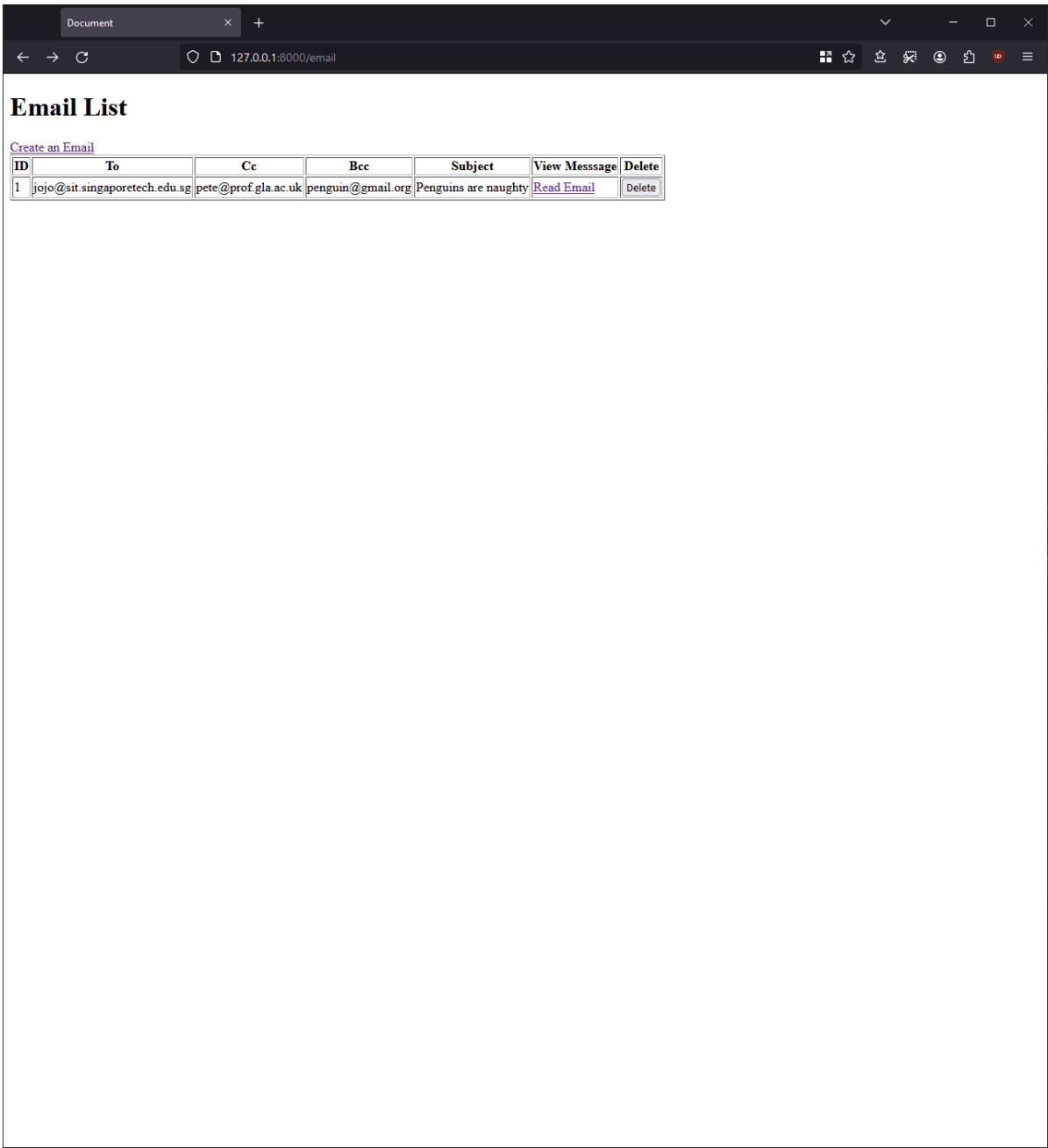
Paste the screenshots in the space below.

This is an individual work.

Please rename this Microsoft Word tutorial+ file to “tuto_plus_05_laravel_[SIT-SID].docx”, replacing [SIT-SID] with your SIT student ID. Submit the file to the corresponding Dropbox before the due date.

=== Happy Programming ===

Screenshots of
The “List Email” interface.



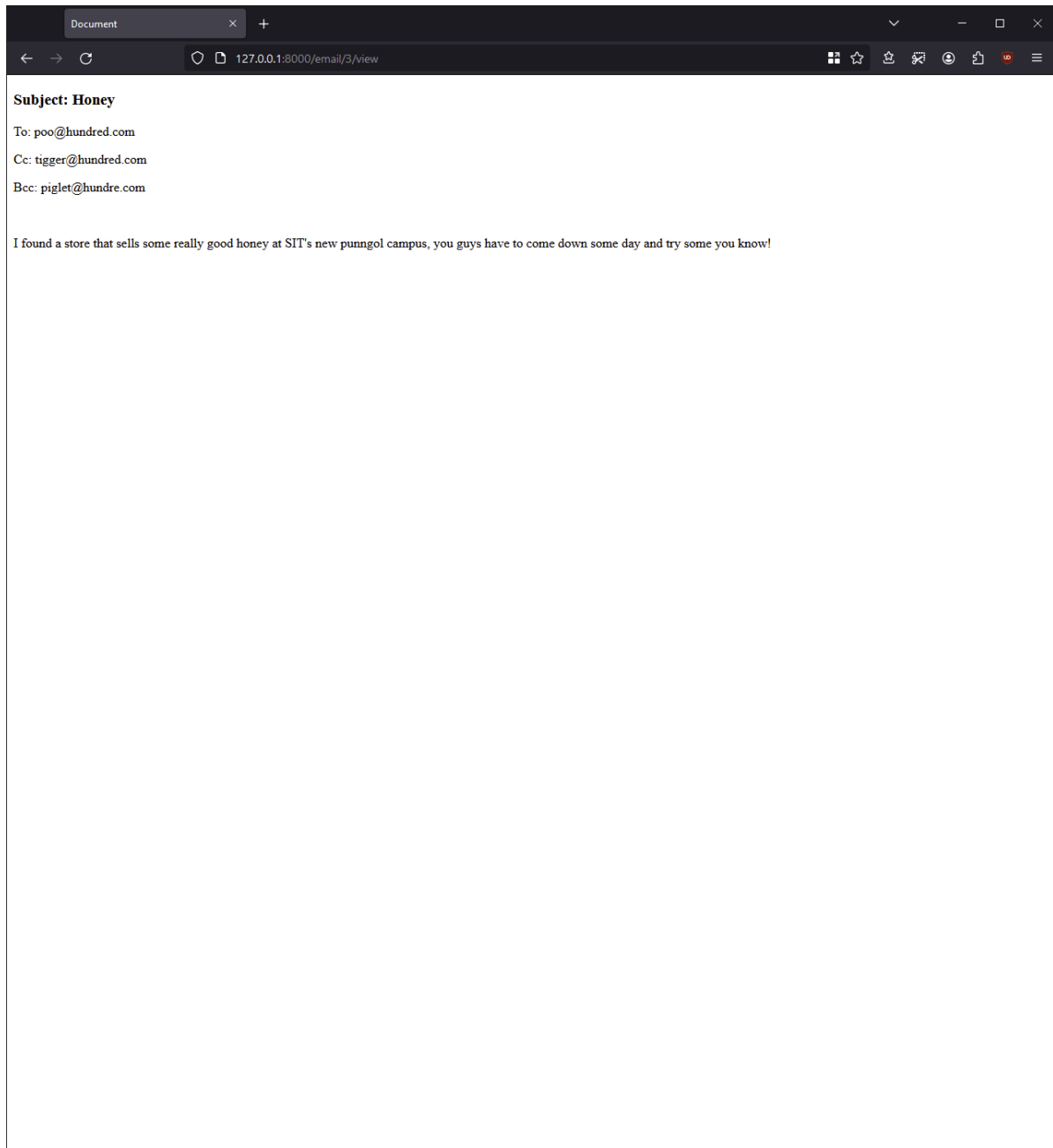
Screenshots of The “Create Email” interface.

The screenshot shows a web browser window with a dark theme. The address bar displays the URL `127.0.0.1:8000/email/create`. The page title is "Write an Email". The form contains the following fields and values:

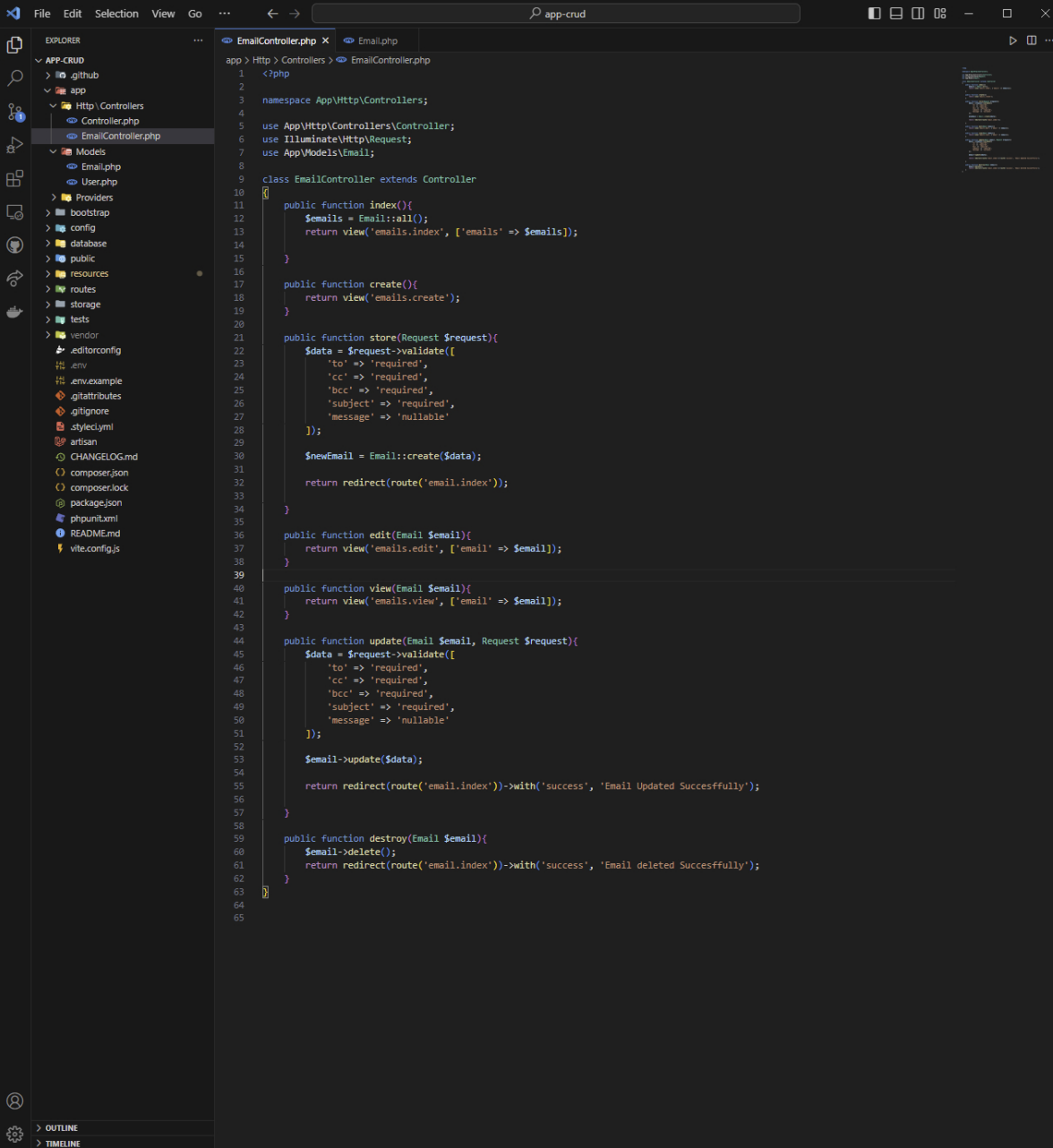
- To: `poo@hundred.com`
- Cc: `tigger@hundred.com`
- Bcc: `piglet@hundre.com`
- Subject: `Honey`
- Message: `ry and try some you know!`

Below the message field is a button labeled "Save a New Email".

Screenshots of The “Read Email” message interface.

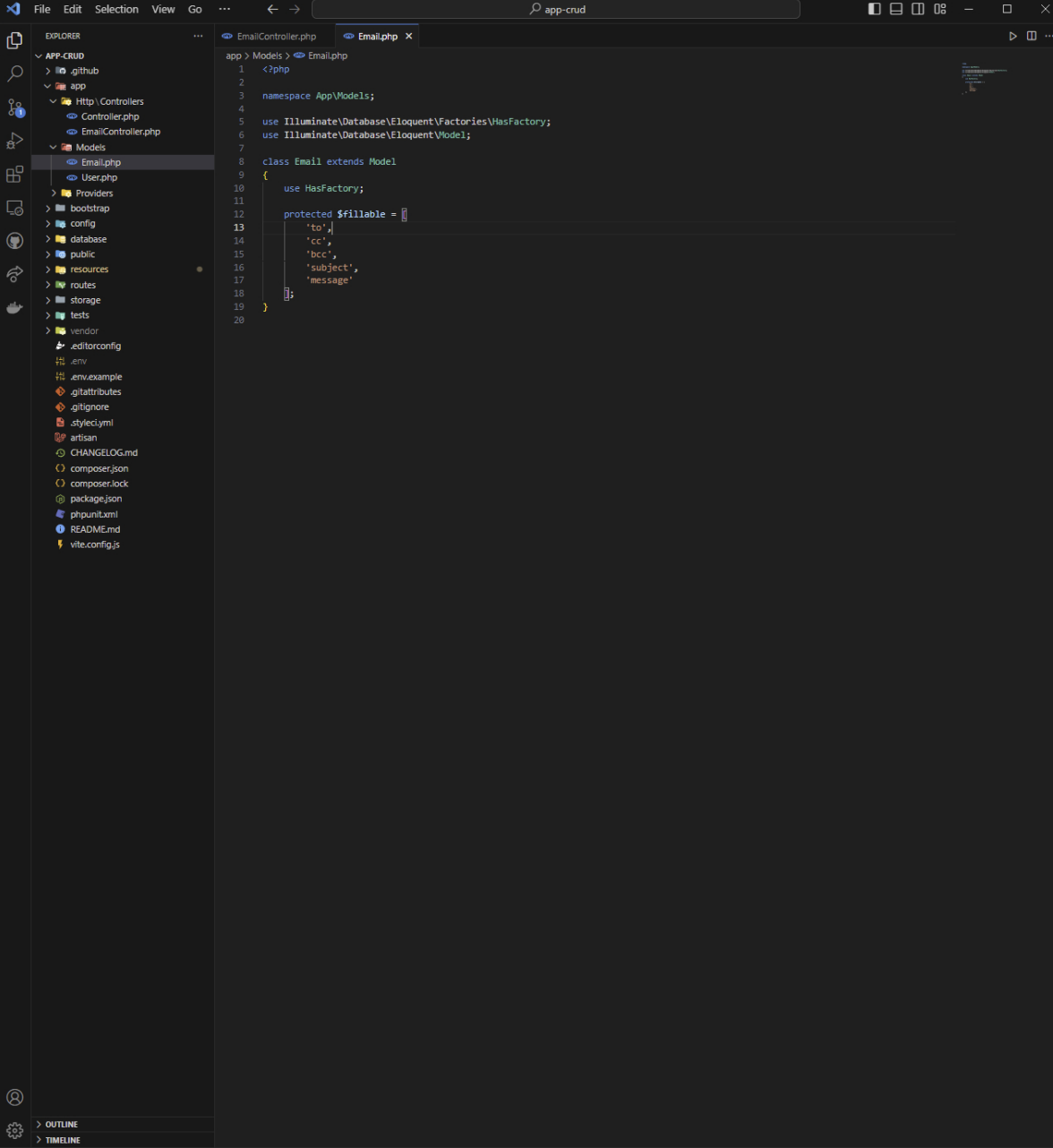


Screenshots of The source code in EmailController.



```
1 <?php
2
3 namespace App\Http\Controllers;
4
5 use App\Http\Controllers\Controller;
6 use Illuminate\Http\Request;
7 use App\Models\Email;
8
9 class EmailController extends Controller
10 {
11     public function index(){
12         $emails = Email::all();
13         return view('emails.index', ['emails' => $emails]);
14     }
15
16     public function create(){
17         return view('emails.create');
18     }
19
20     public function store(Request $request){
21         $data = $request->validate([
22             'to' => 'required',
23             'cc' => 'required',
24             'bcc' => 'required',
25             'subject' => 'required',
26             'message' => 'nullable'
27         ]);
28
29         $newEmail = Email::create($data);
30
31         return redirect(route('email.index'));
32     }
33
34     public function edit(Email $email){
35         return view('emails.edit', ['email' => $email]);
36     }
37
38     public function view(Email $email){
39         return view('emails.view', ['email' => $email]);
40     }
41
42     public function update(Email $email, Request $request){
43         $data = $request->validate([
44             'to' => 'required',
45             'cc' => 'required',
46             'bcc' => 'required',
47             'subject' => 'required',
48             'message' => 'nullable'
49         ]);
50
51         $email->update($data);
52
53         return redirect(route('email.index'))->with('success', 'Email Updated Successfully');
54     }
55
56     public function destroy(Email $email){
57         $email->delete();
58         return redirect(route('email.index'))->with('success', 'Email deleted Successfully');
59     }
60 }
61
62
63
64
65
```

Screenshots of The source code in the “Email” model.



The screenshot shows a code editor with the following structure:

- EXPLORER (Left Panel):** Displays the project file structure. The 'Models' folder is expanded, showing 'Email.php' and 'User.php'.
- Editor (Main Area):** Displays the source code of 'Email.php'. The code is as follows:

```
1 <?php
2
3 namespace App\Models;
4
5 use Illuminate\Database\Eloquent\Factories\HasFactory;
6 use Illuminate\Database\Eloquent\Model;
7
8 class Email extends Model
9 {
10     use HasFactory;
11
12     protected $fillable = [
13         'to',
14         'cc',
15         'bcc',
16         'subject',
17         'message'
18     ];
19 }
20
```
- STATUS BAR (Bottom):** Shows 'Ln 13, Col 14', 'Spaces: 4', 'UTF-8', 'CRLF', and 'PHP'.