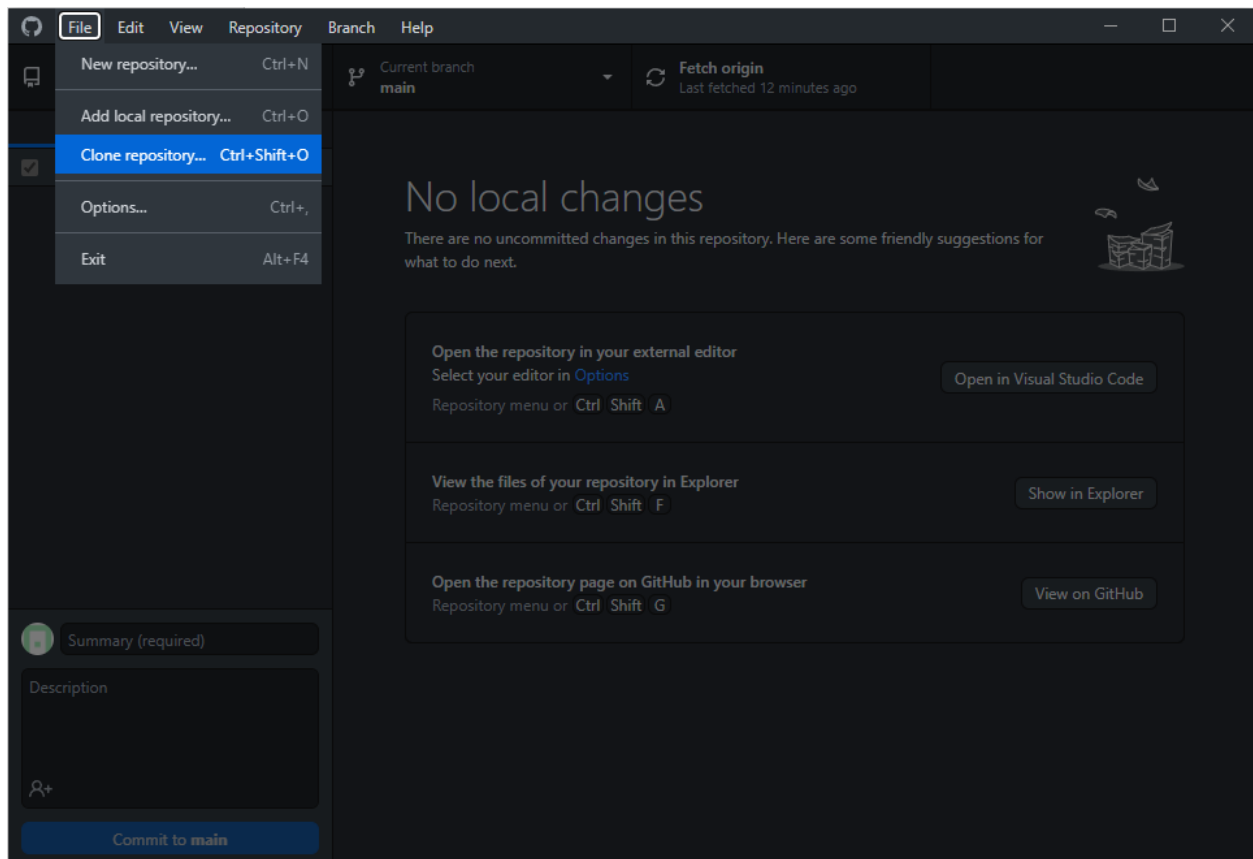


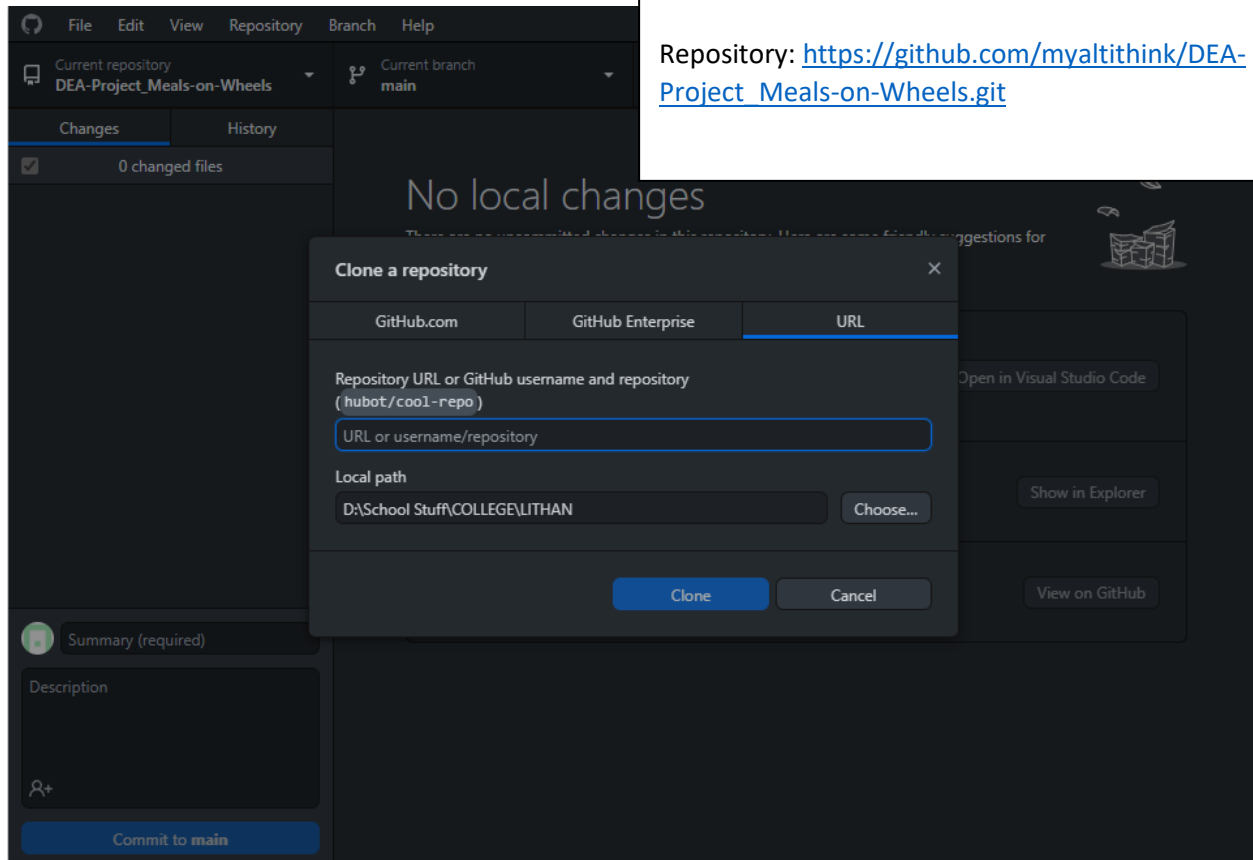
## Meals on Wheels setup

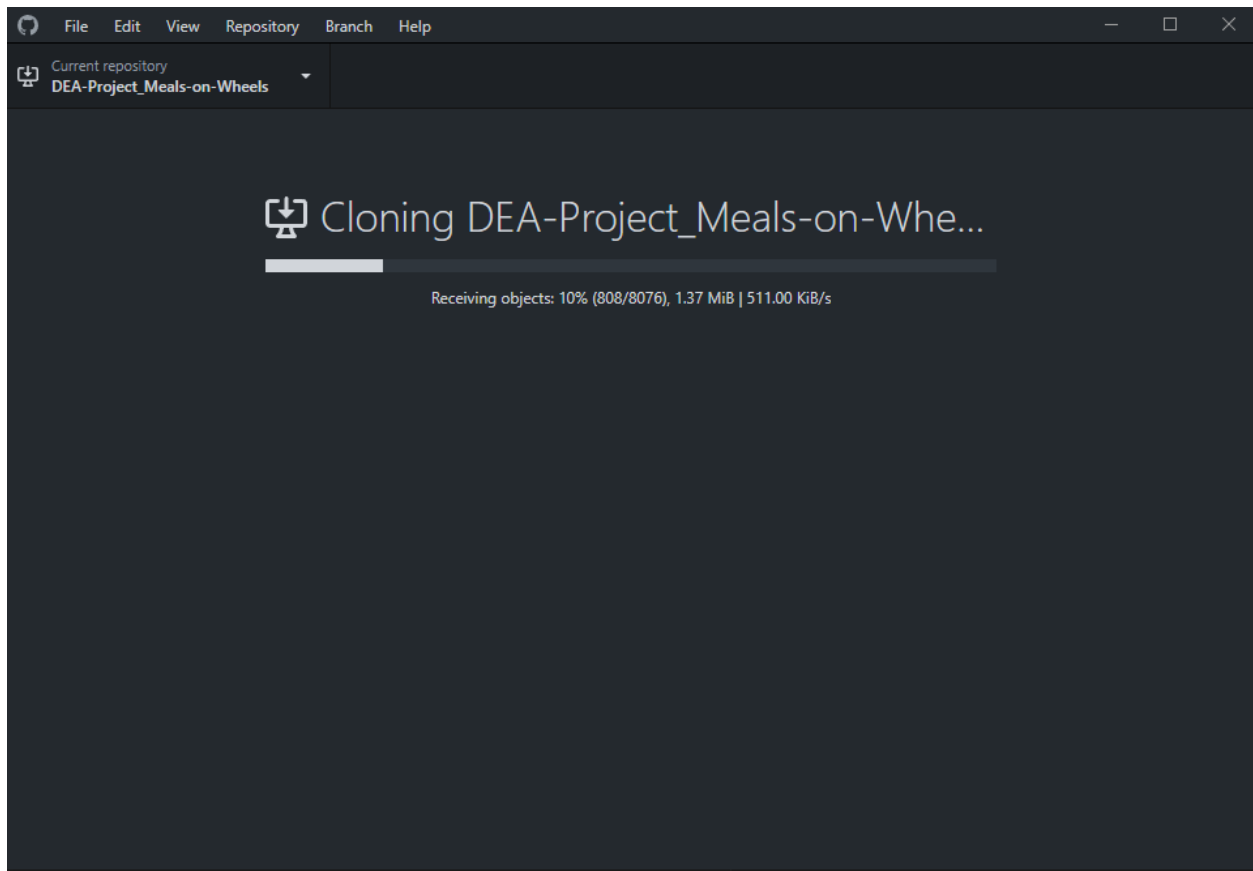
- Install Git kapag wala ka pa
  - Download url <https://git-scm.com/downloads>
- Install GitHub Desktop para di na tayo mag type ng mag type ng command sa cmd
  - Download url <https://desktop.github.com/>
- Cloning a repository



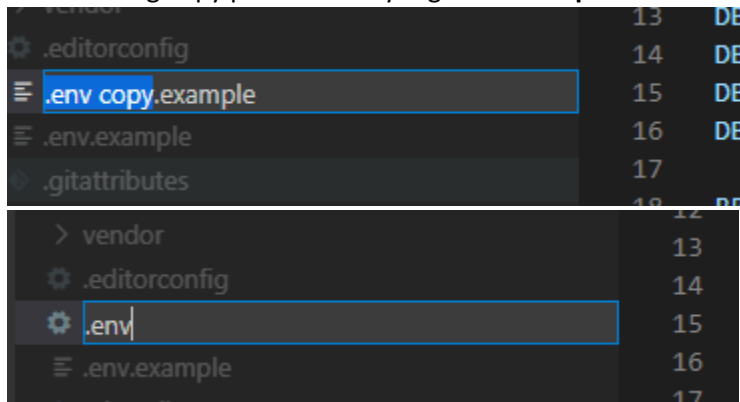
Sa URL tab paste lang yung repository url tas select nalang nag desired location mo kung san mo lalagay

Repository: [https://github.com/myaltithink/DEA-Project\\_Meals-on-Wheels.git](https://github.com/myaltithink/DEA-Project_Meals-on-Wheels.git)

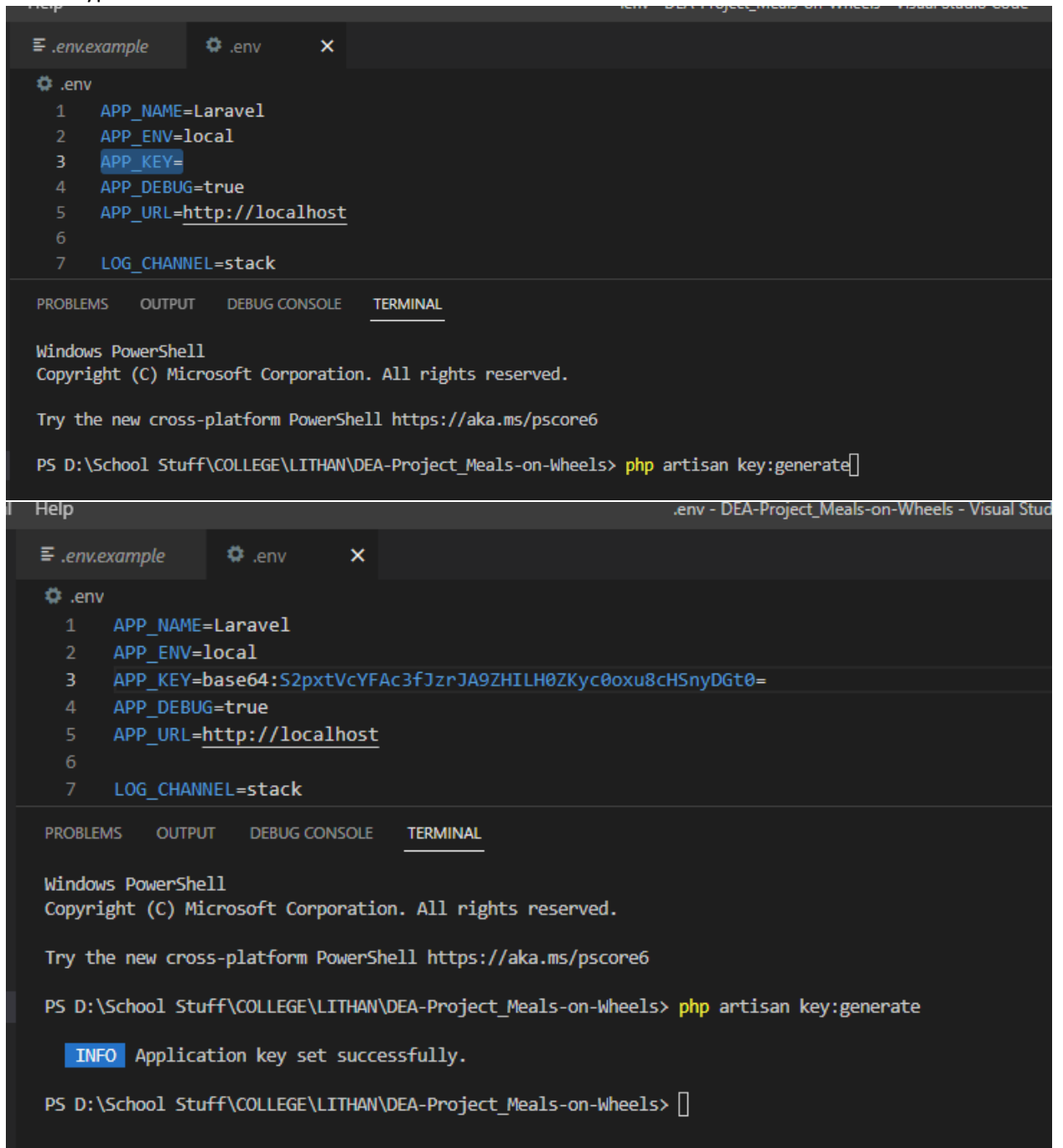




- After cloning copy paste mo na yung **.env.example** na file tas rename mo lang sa **.env**



- Tas type ka ng command na **php artisan key:generate** para ma generate yung application key for encryption



The image consists of two screenshots of a Visual Studio Code editor window. The top screenshot shows the `.env` file with the following content:

```
1 APP_NAME=Laravel
2 APP_ENV=local
3 APP_KEY=
4 APP_DEBUG=true
5 APP_URL=http://localhost
6
7 LOG_CHANNEL=stack
```

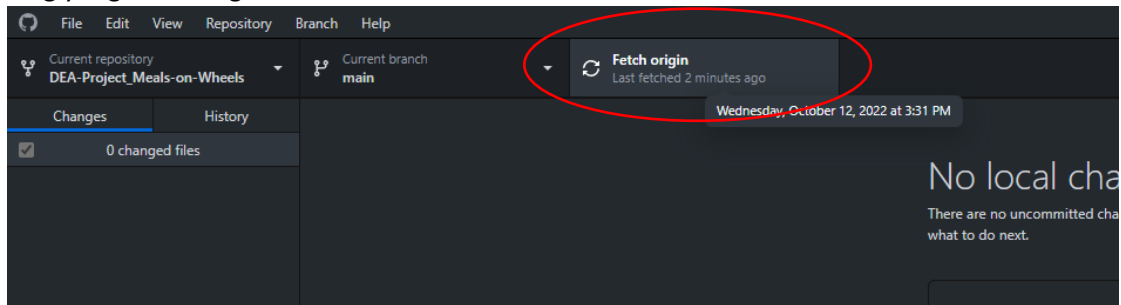
The bottom screenshot shows the same `.env` file, but the `APP_KEY` has been generated and is now a long base64 string:

```
1 APP_NAME=Laravel
2 APP_ENV=local
3 APP_KEY=base64:S2pxtVcYFAC3fJzrJA9ZHILH0ZKyc0oxu8cHSnyDGt0=
4 APP_DEBUG=true
5 APP_URL=http://localhost
6
7 LOG_CHANNEL=stack
```

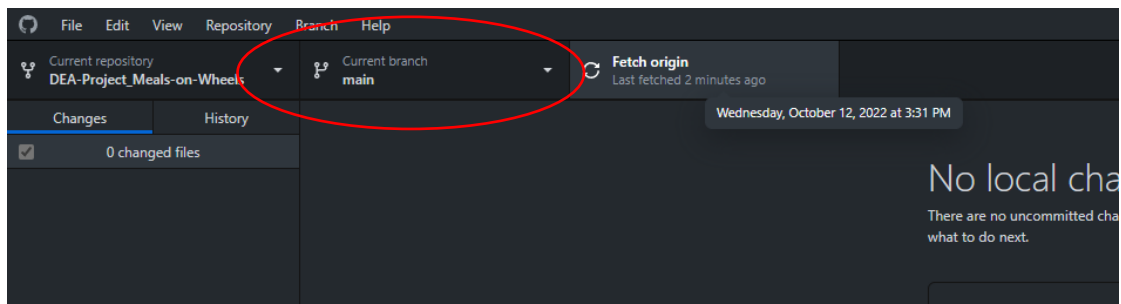
Below the `.env` file, the terminal output is shown. It displays the command `php artisan key:generate` being executed in a PowerShell prompt. The output shows an `INFO` message: `Application key set successfully.`

Pede mo na sya I run through **php artisan serve**

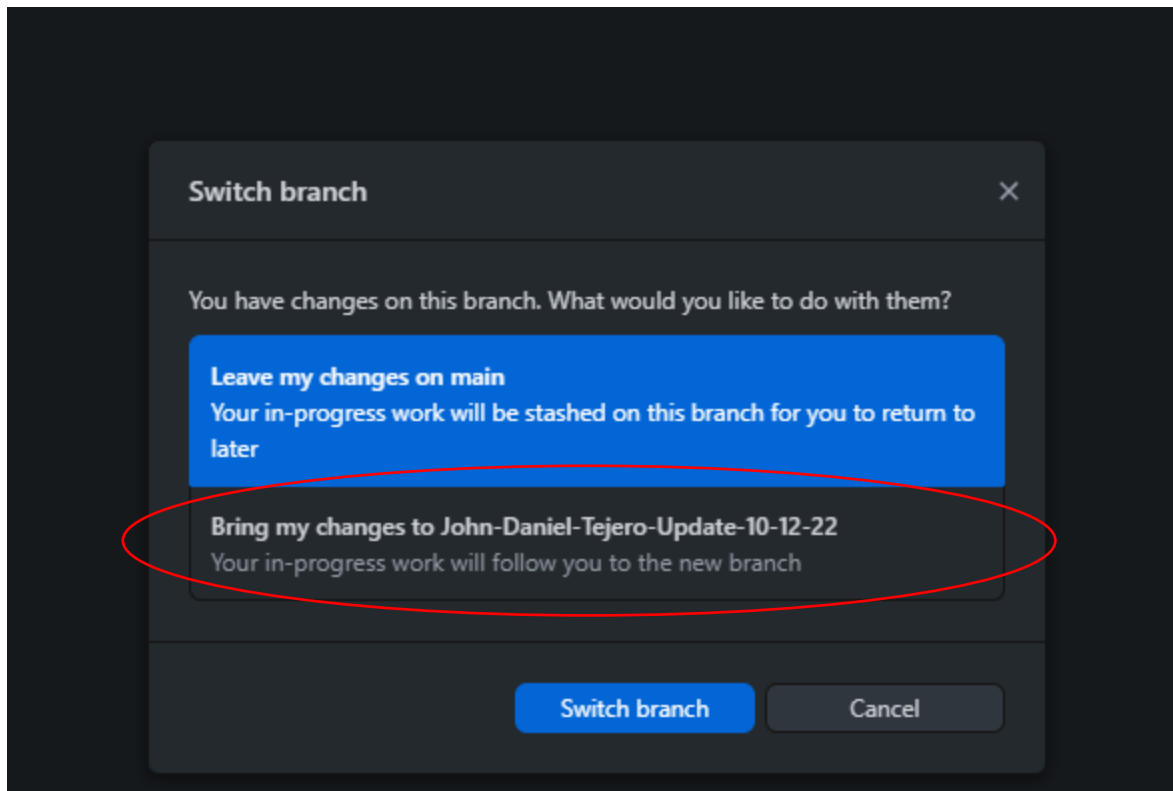
Panigurado kada Commit sa main repository, yung local machine mo is di updated, so click mo lang yung fetch origin sa taas



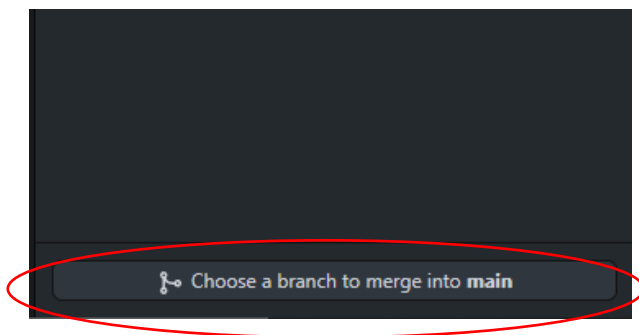
Then after fetching click mo yung current branch, make sure na yan yung branch na ginagamit mo for coding.



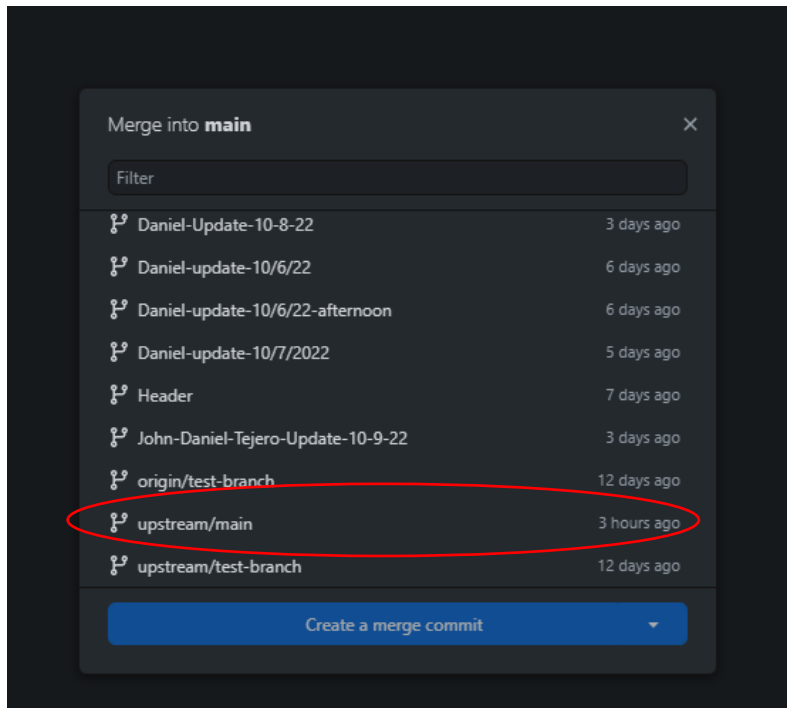
If lilipat ka ng branch, make sure na dadalhin mo yung changes na ginawa mo.



After that click choose branch to merge sa pinakababa



Then sa popup scroll down hanapin mo yung upstream/main, yan yung sa git repository, yung main na branch mo is yung sa local mo. After clicking that click create a merge commit.



After mag merge may lalabas sa screen mo na option to push origin para maupdate yung local copy mo, click mo lang then boom.

## ROLE-BASED Utilities In MEALS ON WHEELS PROJECT by John Daniel Tejero

### ROLE-BASED Rendering in blade

Syempre wala tayong spring security tags dito para ifilter kung alin lang irerender sa frontend. May specific parts lang tayo na gusto natin marender for users depende sa role, example yung sa navbar.

So meron tayong nagawang 4 na custom directives para masatisfy natin yung kulang ng Laravel na kinakailangan nung spring security tags.

Directive	Parameter/s	Description
role/endrole	role('ROLE_HERE') in form of string	Specify one role for rendering a specific portion or section of frontend if true.
hasRoles/endHasRoles	hasRoles(['ROLE_HERE','ROLE_THERE']) in form of array	Specify multiple roles for rendering specific portion or section, all should be satisfied.
HasAnyRole/EndHasAnyRoles	hasAnyRole(['ROLE_HERE','ROLE_THERE']) in form of array	Specify multiple roles for rendering specific portion or section, only one is needed to be satisfied
ExcludeRole/EndExcludeRole	ExcludeRole('ROLE_HERE') in form of string	Specify one role to exclude them from rendering the section or portion of the frontend.

### Example Use Cases:



```

<nav>
  <ul class="nav nav-tabs border border-0">
    @HasAnyRole(['ROLE_MEMBER', 'ROLE_CARETAKER'])
    <li class="nav-item">
      <a class="
        nav-link
        @if (Request::url() == route('meals-list'))
          active
        @endif"
        href="{{route('meals-list')}}"
      >
        Meals
      </a>
    </li>
    <li class="nav-item">
      <a class="
        nav-link
        @if (Request::url() == route('mc-orders'))
          active
        @endif"
        href="{{route('mc-orders')}}"
      >
        Orders
      </a>
    </li>
  </ul>
@EndHasAnyRoles

```

```

@role('ROLE_ADMIN')
<li class="nav-item">
  <a class="
    nav-link
    @if (Request::url() == route('a-prep-orders'))
      active
    @endif"
    href="{{route('a-prep-orders')}}"
  >
    Assign Preparation
  </a>
</li>
<li class="nav-item">
  <a class="
    nav-link
    @if (Request::url() == route('a-del-orders'))
      active
    @endif"
    href="{{route('a-del-orders')}}"
  >
    Assign Delivery
  </a>
</li>
@endrole

```

**ROLE-BASED Filter in middleware**

So sa spring security naalala nyo naman siguro yung prepostenable yung @hasRole('role') sa taas ng controller or something or yung antmatchers. May counterpart si Laravel although di natin magagamit since may custom role thingy tayo through many to many relation. Ang workaround nito is paggamit ng middleware. Pwede nyo sya gamitin sa mismong constructor ng controller, or sa web.php, personally preferred ko if sa web.php mismo since iba't iba ng role access ang bawat function or method ng controller class depende sa business needs.

Middleware	Usage	Description
AuthorizeRole	middleware(['authorizerole:ROLE_ADMIN']);	In authorize role, parameter is taken after the colon ":" and only one is taken
AuthorizeAnyRole	middleware(['anyrole:ROLE_VOLUNTEER_COOK,ROLE_PARTNER']);	In anyrole parameter is taken after colon and separated by comma, make sure there's no space for it to work.

## Example Use Cases

### In web.php

```
//rendering order page for assigning meal to partner/volunteer for admin
Route::get('/assign-orders', [DeliveryManagementController::class, 'ordersForAdminAssignVP'])
    ->name('a-prep-orders')
    ->middleware(['authorizerole:ROLE_ADMIN']);

//rendering order page for assigning meal to partner/volunteer for admin
Route::get('/assign-orders-delivery', [DeliveryManagementController::class, 'ordersForAdminAssignR'])
    ->name('a-del-orders')
    ->middleware(['authorizerole:ROLE_ADMIN']);
```

```
//rendering order page for partner and volunteer for packing
Route::get('/to-pack-orders', [DeliveryManagementController::class, 'ordersForVolunteerPartnerForPacking'])
    ->name('vp-pack-orders')
    ->middleware(['anyrole:ROLE_VOLUNTEER_COOK,ROLE_PARTNER']);
```

### Inside Controller

```
class MealProposalController extends Controller
{
    public function __construct()
    {
        $this->middleware('auth');
        $this->middleware('anyrole:ROLE_VOLUNTEER_COOK,ROLE_PARTNER');
    }

    //viewing meal proposal lists page
```