

Task 3: Scoping task

1. Step one: Creating a model and database table

- a) I would use the command `php artisan make:model CompanyEmail -m` to create model and migration for company email
- b) In the migration file, define fields like `id` (PK), `email`, `companyId` (FK- referencing to `companies.id`) and `type`
- c) Set the `$table` property in the `CompanyEmail` model to `'company_email'` and list the fields in the `$fillable` array

2. Step two: Setting up relationships

- a) Establish a relationship between `Company` and `CompanyEmail` models
- b) Define a One to Many relationship between these models where one `Company` can have multiple `CompanyEmails`
- c) In `Company` model create method `email` which will return `$this->hasMany` with entry parameter `CompanyEmail::class`
- d) in `CompanyEmail` create method `company` which will return `$this->belongsTo` (`Company::class`)

3. Step three: Creating Controller, Service and Repository

- a) Create a new controller which will handle CRUD operations related to company emails
- b) Implement service and repository layers (like in second task)
- c) Write methods in controller to handle CRUD operations, with logic in service layer

4. Step four: Configuring routes

- a) Add new routes in `web.php` and then call methods from Controller

5. Step five: Making changes to frontend

- a) Modify frontend accommodate the new features for managing company emails
- b) Design forms and UI for actions like adding, viewing, editing and deleting company emails