

Contributing to SkillHarbor

1. Check Open Issues: Browse the [issue tracker](link to issue tracker) to find open issues that you can contribute to. Look for bugs to fix, features to implement, or ocumentation improvements to make.
2. Fork the Repository: If you're ready to make changes, fork the SkillHarbor repository to your GitHub account. This will create a copy of the repository that you can freely experiment with.
3. Modify jetstream to cater for the modified fields of the user model. Follow the migration below for guidance.

```
1. Schema::create('users', function (Blueprint $table) {
    $table->id();
    $table->foreignId('section_id')->onDelete('cascade')->nullable();
    $table->foreignId('supervisor_id')->onDelete('cascade')->nullable()->default('1');

    $table->string('first_name');
    $table->string('last_name');
    $table->integer('salary_ref_number')->nullable();
    $table->string('gender')->nullable();
    $table->date('dob')->nullable();
    $table->string('role')->nullable();
    $table->Integer('competency_rating')->default('0');
    $table->string('email')->unique();
    $table->timestamp('email_verified_at')->nullable();
    $table->string('password');
    $table->rememberToken();
    $table->foreignId('current_team_id')->nullable();
    $table->string('profile_photo_path', 2048)->nullable();
    $table->timestamps();
});

Schema::create('password_reset_tokens', function (Blueprint $table) {
    $table->string('email')->primary();
    $table->string('token');
    $table->timestamp('created_at')->nullable();
});

Schema::create('sessions', function (Blueprint $table) {
    $table->string('id')->primary();
    $table->foreignId('user_id')->nullable()->index();
    $table->string('ip_address', 45)->nullable();
    $table->text('user_agent')->nullable();
    $table->longText('payload');
```

```
$table->integer('last_activity')->index();  
});
```

4. We then need to update the Fortify CreateUser Module and updateUserProfileInformation by replacing name with first_name and last_name in the **\App\Actions\Fortify** directory.

Before:

```
public function create(array $input): User  
{  
    Validator::make($input, [  
        'name' => ['required', 'string', 'max:255'],  
        'email' => ['required', 'string', 'email', 'max:255'],  
        'unique:users'],  
        'password' => $this->passwordRules(),  
        'terms' => Jetstream::hasTermsAndPrivacyPolicyFeature() ?  
        ['accepted', 'required'] : '',  
    ]->validate();  
  
    return User::create([  
        'name' => $input['name'],  
        'email' => $input['email'],  
        'password' => Hash::make($input['password']),  
    ]);  
}
```

After:

```
public function create(array $input): User  
{  
    Validator::make($input, [  
        'first_name' => ['required', 'string', 'max:255'],  
        'last_name' => ['required', 'string', 'max:255'],  
        'email' => ['required', 'string', 'email', 'max:255'],  
        'unique:users'],  
        'password' => $this->passwordRules(),  
        'terms' => Jetstream::hasTermsAndPrivacyPolicyFeature() ?  
        ['accepted', 'required'] : '',  
    ]->validate();  
  
    return User::create([
```

```

        'first_name' => $input['first_name'],
        'last_name' => $input['last_name'],
        'email' => $input['email'],
        'password' => Hash::make($input['password']),
    ]);
}

```

5. Then modify the blade template for registration in `\resources\views\auth\register.blade.php`
Before:

```

<div>
    <x-label for="name" value="{{ __('Name') }}" />
    <x-input id="name" class="block mt-1 w-full"
type="text" name="name" :value="old('name')" required autofocus
autocomplete="name" />
</div>

```

After:

```

<div>
    <x-label for="first_name" value="{{ __('First Name')
}}" />
    <x-input id="first_name" class="block mt-1 w-full"
type="text" name="first_name" :value="old('first_name')" required
autofocus autocomplete="first_name" />
</div>
<div>
    <x-label for="last_name" value="{{ __('Last Name') }}"
/>
    <x-input id="last_name" class="block mt-1 w-full"
type="text" name="last_name" :value="old('last_name')" required
autofocus autocomplete="last_name" />
</div>

```

6. Make Your Changes: Make the necessary changes to address the issue you're working on. Follow the coding standards and style guidelines outlined in the contribution guidelines.
7. Submit a Pull Request: Once you're happy with your changes, submit a pull request to the main SkillHarbor repository. Provide a clear description of your changes and reference any related issues or pull requests.
8. Review and Iterate: Your pull request will be reviewed by the project maintainers. They may provide feedback or request additional changes. Iterate on your changes as needed until they are approved and merged into the main repository.