

Guidelines for Getting Started with Development:

Step-0: Clone our project repository from [here](#).

1. Installing the necessary software packages:

- a. The Ruby version used for the development of our application is **3.3.2**. If your Ruby version is different, we recommend matching ours to avoid any conflicts with dependencies. You may follow the steps given [here](#) for installing Ruby.
- b. If rails is not present already, to install it, do this: `gem install rails`
- c. If bundler is not present already, to install it, do this: `gem install bundler`
- d. If heroku command line (CLI) tool is not present already, to install it, [follow the instructions here](#).
- e. Check the ruby, rails, bundler and heroku CLI versions using the following:
`ruby -v && rails -v && bundle -v && heroku -v`
- f. Install `pkg-config` using this command: `sudo apt-get update && sudo apt-get install pkg-config`

2. Running migrations and creating the database table for development:

- a. The migration files are already present under `db/migrate` folder. So, no need to create any new migration files.
- b. To apply the migration and create the database table for development, simply run `rails db:migrate`.

3. Verification of Routes and Controller Actions

To ensure the application routes are correctly configured and match the intended controller actions, follow these steps:

Run Rails Routes Command: Use the following command in your terminal to list all defined routes: `rails routes`

i) This command outputs a table of routes, their HTTP methods, and the corresponding controller actions, something like this -

```

sakashamv22@MSI:~/TAMU_Sem1/SE/csce-606-autograder-frontend$ rails routes

```

	Prefix	Verb	URI Pattern	Controller#Action
	root	GET	/	pages#home
	pages_home	GET	/pages/home{.:format}	pages#home
	users	GET	/users{.:format}	users#index
	user	GET	/users/:id{.:format}	users#show
	assignments	GET	/assignments{.:format}	assignments#index
	update_user_assignments	POST	/users/:id/update_assignments{.:format}	users#update_assign
ments	update_assignment_users	POST	/assignments/:id/update_users{.:format}	assignments#update_
users		POST	/assignments/:assignment_id/update_test_order{.:format}	assignments#update_
order	assignment_tests	GET	/assignments/:assignment_id/tests{.:format}	tests#index
		POST	/assignments/:assignment_id/tests{.:format}	tests#create
	new_assignment_test	GET	/assignments/:assignment_id/tests/new{.:format}	tests#new
	edit_assignment_test	GET	/assignments/:assignment_id/tests/:id/edit{.:format}	tests#edit
	assignment_test	GET	/assignments/:assignment_id/tests/:id{.:format}	tests#show

ii) Verify Controller Actions:

- Match the routes in the output with their corresponding controller actions.
- Confirm that each route is implemented as an action in the appropriate controller and behaves as expected.

4. Description of Models and Controllers

Models

- ApplicationRecord:**
 - Base class for all models.
 - Provides shared functionality for ActiveRecord models.
- Assignment:**
 - Represents assignments in the application.
- Permission:**
 - Represents permissions associated with users or assignments.
 - Used for managing access control.
- TestGrouping:**
 - Represents groupings of tests within assignments.
 - Used for organizing related tests.
- Test:**
 - Represents individual tests associated with assignments or groupings.
- User:**
 - Represents an application user.

Controllers

- ApplicationController:**

- Base controller for all other controllers.
- Handles shared functionality such as authentication and error handling.
- 2. **AssignmentsController:**
 - Manages operations related to assignments.
 - Includes actions for viewing assignments, updating users assigned to an assignment, searching assignments, managing directory structures.
- 3. **PagesController:**
 - Manages redirection to Assignments page for logged in users.
 - Includes actions for redirecting to assignments for a logged in user.
- 4. **SessionsController:**
 - Manages user sessions.
 - Includes actions for logging in via GitHub, handling login failures, logging out.
 - Verifies organizational membership during login.
- 5. **TestGroupingsController:**
 - Manages groupings of tests.
 - Includes actions for creating and deleting test groupings.
- 6. **TestsController:**
 - Manages individual tests.
 - Includes actions for configuring test metadata.
- 7. **UsersController:**
 - Handles user-related operations.
 - Includes actions for viewing users, managing user assignments.

5. Gem files already done need bundle install

- a. All the Gem files needed are already present in the **Gemfile**. So, no need to add other gem files.
- b. To install all the Gem files, simply run **bundle install**.

6. Adding developer as our organization member in Github

- a. To run **rails server** locally, values of **GITHUB_CLIENT_ID**, **GITHUB_CLIENT_SECRET** and **GITHUB_ACCESS_TOKEN** need to be added in **config/local_env.yml** file.
- b. Attach these three lines to **config/local_env.yml**:


```
GITHUB_CLIENT_ID: "<GITHUB_CLIENT_ID_OAUTH>"
GITHUB_CLIENT_SECRET: "<GITHUB_CLIENT_SECRET_OAUTH>"
```

7. Setting up OAuth Application

You can create and register an OAuth app under your personal account or under any organization you have administrative access to. Here, you should open it under the organization.

1. In the upper-right corner of any page on GitHub, click your profile photo, then click Settings.
2. In the left sidebar, click Developer settings.
3. In the left sidebar, click OAuth apps.
4. Click New OAuth App. (If you haven't created an app before, this button will say, Register a new application.)
5. In "Application name", type the name of your app.
6. In "Homepage URL", type the full URL to your local app's website (If you have deployed in Heroku, provide that URL). For example:
 - a. <http://127.0.0.1:3000/>
7. Optionally, in "Application description", type a description of your app that users will see.
8. In "Authorization callback URL", type the callback URL of your local app.
 - a. For example: <http://127.0.0.1:3000/auth/github/callback>
 - b. NOTE: Ensure that the callback URL is configured in `routes.rb`
9. If your OAuth app will use the device flow to identify and authorize users, click Enable Device Flow. For more information about the device flow, see "[Authorizing OAuth apps](#)."
10. Click Register application.

After registering the OAuth you will get a `GITHUB_CLIENT_ID`, and `GITHUB_CLIENT_SECRET` which needs to be added to `config/local_env.yml` like specified in the previous step.