

UIT2602 WEB PROGRAMMING
Exercise 4 - To-Do List And Blog Post Using Rails

Name: S. Selcia
Reg. No.: 3122215002098
Section: IT B

TO-DO LIST

1. Aim:

The aim of this experiment is to create a web-based ToDo application using Ruby on Rails. The application should allow users to manage their tasks by providing features for creating, updating, and deleting tasks.

2. Required Web Tools and Methodology:

- Ruby on Rails: The web framework will be used to develop the application.
- Web Browser: To test the functionality of the website.

3. Implementation Procedure:

- a. Install Ruby on Rails and set up a new Rails application.
- b. Generate a scaffold for the ToDo resource with title and description attributes.
- c. Run database migrations to create the necessary database tables.
- d. Start the Rails server and test the application locally.
- e. Perform CRUD operations on the new application which was created for managing the tasks.

4. Output Screenshots:

❖ The `'rails new'` command is used to create a new Ruby on Rails application

```
C:\Users\Dell>rails new to_do_app
create
create  README.md
create  Rakefile
create  .ruby-version
create  config.ru
create  .gitignore
create  .gitattributes
create  Gemfile
run git init from "."
Initialized empty Git repository in C:/Users/Dell/to_do_app/.git/
create  app
create  app/assets/config/manifest.js
create  app/assets/stylesheets/application.css
create  app/channels/application_cable/channel.rb
create  app/channels/application_cable/connection.rb
create  app/controllers/application_controller.rb
create  app/helpers/application_helper.rb
create  app/jobs/application_job.rb
create  app/mailers/application_mailer.rb
create  app/models/application_record.rb
create  app/views/layouts/application.html.erb
create  app/views/layouts/mailer.html.erb
create  app/views/layouts/mailer.text.erb
create  app/assets/images
create  app/assets/images/.keep
create  app/controllers/concerns/.keep
```

- ❖ The `'rails generate scaffold'` command creates a set of files for a model, views, and controller for a specified resource, providing a full CRUD (Create, Read, Update, Delete) interface for managing that resource.

```
C:\Users\Dell>cd to_do_app

C:\Users\Dell\to_do_app>rails generate scaffold ToDo title:string description:text
   invoke  active_record
   create  db/migrate/20240313160152_create_to_dos.rb
   create  app/models/to_do.rb
   invoke  test_unit
   create  test/models/to_do_test.rb
   create  test/fixtures/to_dos.yml
   invoke  resource_route
   route   resources :to_dos
   invoke  scaffold_controller
   create  app/controllers/to_dos_controller.rb
   invoke  erb
   create  app/views/to_dos
   create  app/views/to_dos/index.html.erb
   create  app/views/to_dos/edit.html.erb
   create  app/views/to_dos/show.html.erb
   create  app/views/to_dos/new.html.erb
   create  app/views/to_dos/_form.html.erb
   create  app/views/to_dos/_to_do.html.erb
   invoke  resource_route
   invoke  test_unit
   create  test/controllers/to_dos_controller_test.rb
   create  test/system/to_dos_test.rb
   invoke  helper
   create  app/helpers/to_dos_helper.rb
   invoke  test_unit
   invoke  jbuilder
   create  app/views/to_dos/index.json.jbuilder
   create  app/views/to_dos/show.json.jbuilder
   create  app/views/to_dos/_to_do.json.jbuilder
```

- ❖ The `'rails db:migrate'` runs pending database migrations to update the database schema.


```
C:\Users\Dell\to_do_app>rails db:migrate
== 20240313160152 CreateToDos: migrating =====
-- create_table(:to_dos)
   -> 0.0055s
== 20240313160152 CreateToDos: migrated (0.0064s) =====
```

- ❖ The `'rails server'` starts the Rails development server, allowing you to run your Rails application locally. It serves your application so that you can access it through a web browser at `'http://localhost:3000'`.

```
C:\Users\Dell\to_do_app>rails server
=> Booting Puma
=> Rails 7.1.3.2 application starting in development
=> Run 'bin/rails server --help' for more startup options
*** SIGUSR2 not implemented, signal based restart unavailable!
*** SIGUSR1 not implemented, signal based restart unavailable!
*** SIGHUP not implemented, signal based logs reopening unavailable!
Puma starting in single mode...
* Puma version: 6.4.2 (ruby 3.2.3-p157) ("The Eagle of Durango")
* Min threads: 5
* Max threads: 5
* Environment: development
* PID: 12680
* Listening on http://[::1]:3000
* Listening on http://127.0.0.1:3000
Use Ctrl-C to stop
Started GET "/" for 127.0.0.1 at 2024-03-13 21:35:14 +0530
ActiveRecord::SchemaMigration Load (0.3ms)  SELECT "schema_migrations"."version" FROM "schema_migrations" ORDER BY "schema_migrations"."version" ASC
Processing by Rails::WelcomeController#index as HTML
  Rendering C:/Ruby32-x64/lib/ruby/gems/3.2.0/gems/railties-7.1.3.2/lib/rails/templates/rails/welcome/index.html.erb
  Rendered C:/Ruby32-x64/lib/ruby/gems/3.2.0/gems/railties-7.1.3.2/lib/rails/templates/rails/welcome/index.html.erb (Duration: 3.1ms | Allocations: 376)
Completed 200 OK in 497ms (Views: 185.9ms | ActiveRecord: 0.8ms | Allocations: 8813)

Started GET "/" for 127.0.0.1 at 2024-03-13 21:35:38 +0530
Processing by Rails::WelcomeController#index as HTML
  Rendering C:/Ruby32-x64/lib/ruby/gems/3.2.0/gems/railties-7.1.3.2/lib/rails/templates/rails/welcome/index.html.erb
  Rendered C:/Ruby32-x64/lib/ruby/gems/3.2.0/gems/railties-7.1.3.2/lib/rails/templates/rails/welcome/index.html.erb (Duration: 0.1ms | Allocations: 13)
```

- ❖ Once we enter the browser with http://127.0.0.1:3000/to_dos, we get this



← → ↻ 🔍 ⓘ 127.0.0.1:3000/to_dos

To dos

[New to do](#)

- ❖ Create: Allows users to add new records to the database.



← ↻ 🔍 ⓘ 127.0.0.1:3000/to_dos/new

New to do

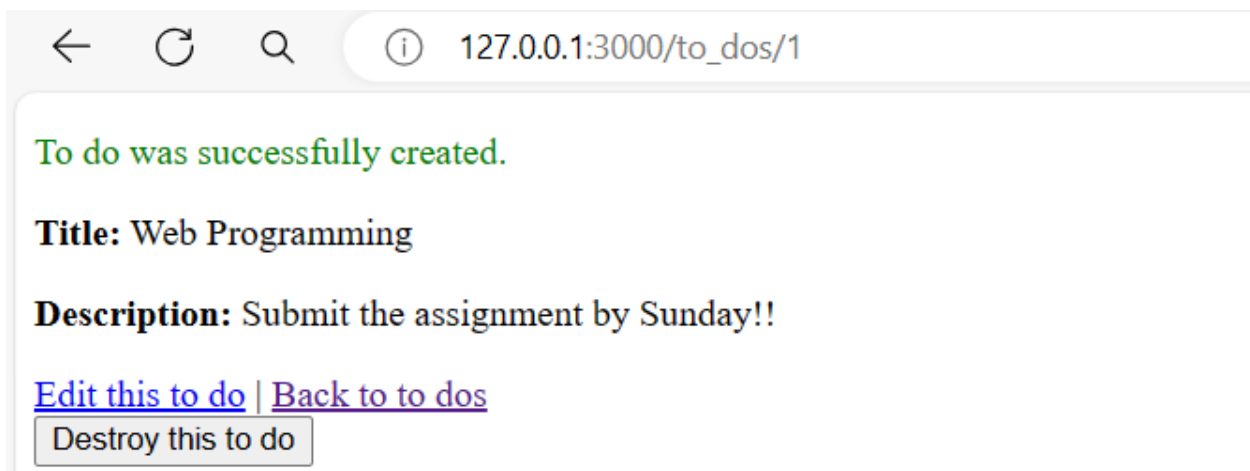
Title
Web Programming

Description
Submit the assignment by Sunday!!

Create To do

[Back to to dos](#)

- ❖ Read: Enables users to view existing records in the database.



← ↻ 🔍 ⓘ 127.0.0.1:3000/to_dos/1

To do was successfully created.





Title: Web Programming

Description: Submit the assignment by Sunday!!

[Edit this to do](#) | [Back to to dos](#)

Destroy this to do


- ❖ Update: Allows users to modify existing records in the database.

 127.0.0.1:3000/to_dos/1/edit

Editing to do

Title





Description



Update To do

[Show this to do](#) | [Back to to dos](#)

- ❖ Read: Enables users to view existing records in the database.

 127.0.0.1:3000/to_dos/1

To do was successfully updated.

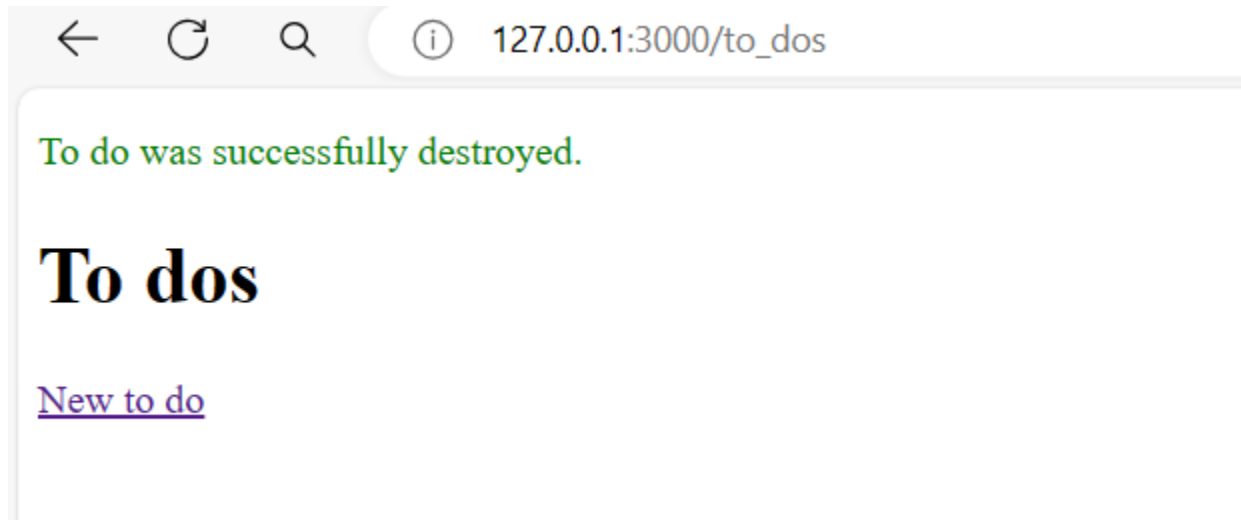
Title: AR/VR Exam

Description: CAT 1 begins on Monday

[Edit this to do](#) | [Back to to dos](#)

Destroy this to do

- ❖ Delete: Allows users to remove records from the database.



5. Conclusion:

In conclusion, this experiment demonstrated the process of creating a web-based ToDo application using Ruby on Rails. By following the steps outlined above, a functional ToDo application was developed with features for managing tasks. This experiment highlights the power and simplicity of Ruby on Rails for developing web applications.

BLOG POST

1. Aim:

The aim of this experiment is to create a web-based application using Ruby on Rails. The application should allow users to create, update and delete blog posts.

2. Required Web Tools and Methodology:

- Ruby on Rails: The web framework will be used to develop the application.
- Web Browser: To test the functionality of the website.

3. Implementation Procedure:

- a. Install Ruby on Rails and set up a new Rails application.
- b. Generate a scaffold for the blogpost resource with title and content attributes.
- c. Run database migrations to create the necessary database tables.
- d. Start the Rails server and test the application locally.

- e. Perform CRUD operations on the new application which was created for managing the posts.

4. Output Screenshots:

- ❖ The `'rails new'` command is used to create a new Ruby on Rails application

```
C:\Users\Dell>rails new blogpost
create
create  README.md
create  Rakefile
create  .ruby-version
create  config.ru
create  .gitignore
create  .gitattributes
create  Gemfile
run     git init from "."
Initialized empty Git repository in C:/Users/Dell/blogpost/.git/
create  app
create  app/assets/config/manifest.js
create  app/assets/stylesheets/application.css
create  app/channels/application_cable/channel.rb
create  app/channels/application_cable/connection.rb
create  app/controllers/application_controller.rb
create  app/helpers/application_helper.rb
create  app/jobs/application_job.rb
create  app/mailers/application_mailer.rb
create  app/models/application_record.rb
create  app/views/layouts/application.html.erb
create  app/views/layouts/mailer.html.erb
create  app/views/layouts/mailer.text.erb
create  app/assets/images
create  app/assets/images/.keep
create  app/controllers/concerns/.keep
create  app/models/concerns/.keep
```

- ❖ The `'rails generate scaffold'` command creates a set of files for a model, views, and controller for a specified resource, providing a full CRUD (Create, Read, Update, Delete) interface for managing that resource.

```
C:\Users\Dell>cd blogpost

C:\Users\Dell\blogpost>rails generate scaffold Post title:string content:text
invoke  active_record
create  db/migrate/20240313174026_create_posts.rb
create  app/models/post.rb
invoke  test_unit
create  test/models/post_test.rb
create  test/fixtures/posts.yml
invoke  resource_route
route   resources :posts
invoke  scaffold_controller
create  app/controllers/posts_controller.rb
invoke  erb
create  app/views/posts
create  app/views/posts/index.html.erb
create  app/views/posts/edit.html.erb
create  app/views/posts/show.html.erb
create  app/views/posts/new.html.erb
create  app/views/posts/_form.html.erb
create  app/views/posts/_post.html.erb
invoke  resource_route
invoke  test_unit
create  test/controllers/posts_controller_test.rb
create  test/system/posts_test.rb
invoke  helper
```

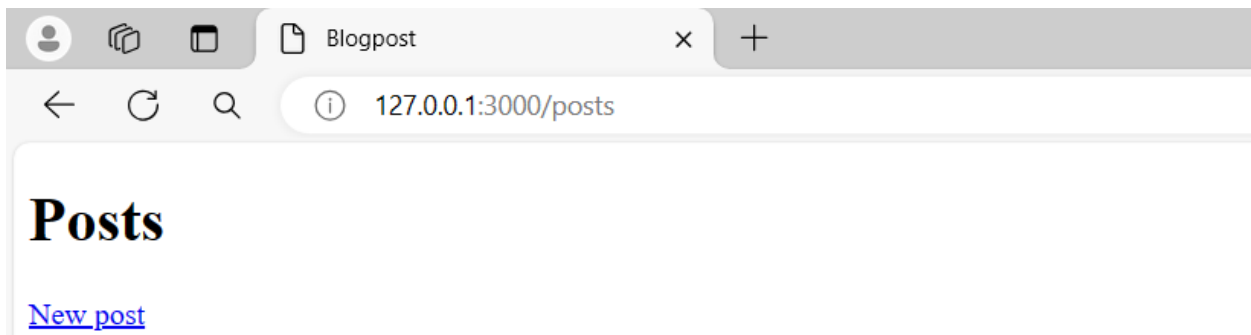
- ❖ The `'rails db:migrate'` runs pending database migrations to update the database schema.

```
C:\Users\Dell\blogpost>rails db:migrate
== 20240313174026 CreatePosts: migrating =====
-- create_table(:posts)
   -> 0.0150s
== 20240313174026 CreatePosts: migrated (0.0155s) =====
```

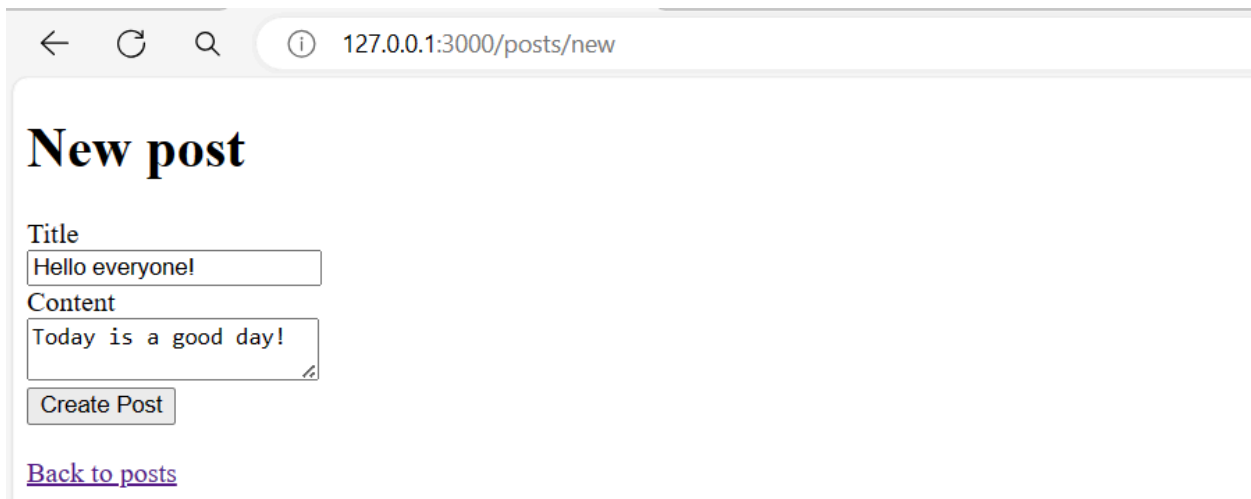
- ❖ The `'rails server'` starts the Rails development server, allowing you to run your Rails application locally. It serves your application so that you can access it through a web browser at `'http://localhost:3000'`.

```
C:\Users\Dell\blogpost>rails server
=> Booting Puma
=> Rails 7.1.3.2 application starting in development
=> Run 'bin/rails server --help' for more startup options
*** SIGUSR2 not implemented, signal based restart unavailable!
*** SIGUSR1 not implemented, signal based restart unavailable!
*** SIGHUP not implemented, signal based logs reopening unavailable!
Puma starting in single mode...
* Puma version: 6.4.2 (ruby 3.2.3-p157) ("The Eagle of Durango")
* Min threads: 5
* Max threads: 5
* Environment: development
* PID: 8620
* Listening on http://[::1]:3000
* Listening on http://127.0.0.1:3000
```

- ❖ Once we enter the browser with http://127.0.0.1:3000/to_dos, we get this



- ❖ Create: Allows users to add new records to the database.



- ❖ Read: Enables users to view existing records in the database.

← ↻ 🔍 ⓘ 127.0.0.1:3000/posts/1

Post was successfully created.

Title: Hello everyone!

Content: Today is a good day!

[Edit this post](#) | [Back to posts](#)

Destroy this post

❖ Update: Allows users to modify existing records in the database.

← ↻ 🔍 ⓘ 127.0.0.1:3000/posts/1/edit

Editing post

Title

Content

Update Post

[Show this post](#) | [Back to posts](#)

❖ Read: Enables users to view existing records in the database.

← ↻ 🔍 ⓘ 127.0.0.1:3000/posts/1

Post was successfully updated.

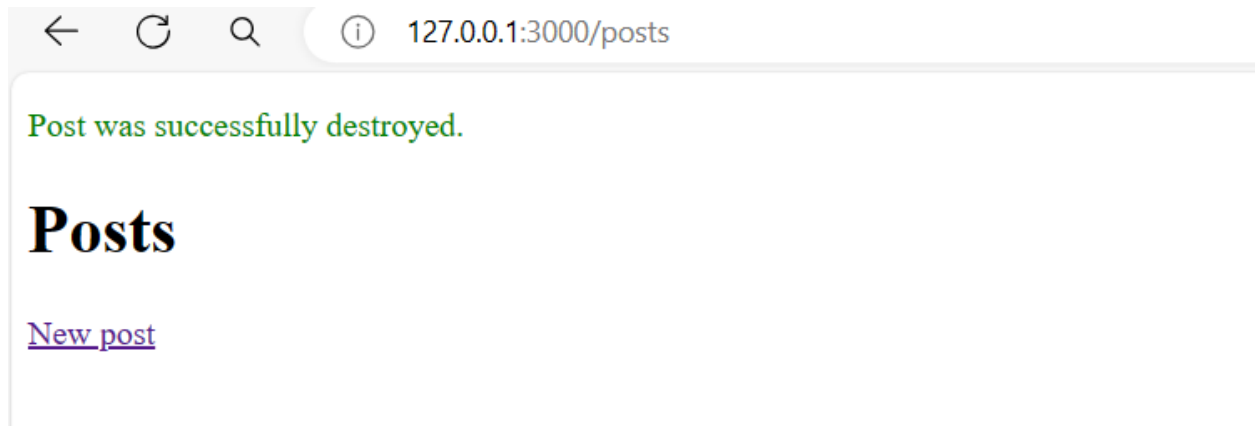
Title: Hi this is Selcia

Content: I am doing my assignments..

[Edit this post](#) | [Back to posts](#)

Destroy this post

- ❖ Delete: Allows users to remove records from the database.



6. Conclusion:

This experiment demonstrated the process of creating a basic blog post functionality in a Ruby on Rails application. By following the steps outlined above, a functional blog application was developed that allows users to manage blog posts. Rails' scaffold generator which made it easy to implement CRUD operations for the blog posts.