

Homework 5: MVC Controller

For this homework, you will build upon the models from the previous homework to create an actual working web app – mainly by adding controllers and views.

You will do this homework as a team; however, each member of your team will be responsible for the completion of a particular task.

Like the last homework, each team member will choose one task from the list below to complete. All team members must do a different task. If your team has only four members, then ignore Task 5. IMPORTANT: So I can tell who did what, list each team member’s username in the project’s README file along with the task that he/she completed.

How to submit your team’s work

Before you can submit, all team members must have merged their code into the master branch and pushed the updates to GitHub. If a team member does not complete his/her work on time, you may submit without his/her contribution.

To submit your team’s work, you must “tag” the current commit in the master branch:

```
$ git tag -a hw5v1 -m 'Tagged Homework 5 submission (version 1)'  
$ git push origin --tags
```

To grade your work, I will check out the appropriate tag, and run it on my machine.

Note that if for some reason you need to update your submission, simply repeat the tagging process, but increment the version number (e.g., hw5v2, hw5v3, hw5v4, etc.).

The Tasks

For this homework, you will create controllers and views that enable users to manipulate instances of the model classes you created last homework. You should work on the task below with the same task number you chose last time. (If you think you should change task numbers, get my permission first.)

In particular, you must:

- Create 5 different seed-data entries for your model object (i.e., in `db/seeds.rb`). Be sure to use the `create!` method (with a `!`) to create them, so you will get error messages if a validation error occurs.
- Create a controller class that corresponds to model. For example, if your model class was `Dog`, then your controller should be `DogsController` (note that “dog” is singular in the model class name and plural in the controller name).
- Add controller actions and views as per the task-specific instructions below.
- On the home page (`static_pages/home.html.erb`), modify the link you originally created in Homework 3 such that it now links to your index page (every task below involves creating an index page).

As additional constraints, you must:

- Create your controller actions and view ERBs by hand (i.e., not using scaffold).
- Along those lines, your controller actions may not mention “json”, nor may you use the `resource` method in `config/routes.rb`.
- You must implement only the actions/views requested (no extras).

Task 1

Create controller actions/views to manage this model class:

OwnershipRecord
year_acquired : integer year_relinquished : integer purchase_price : integer

Controller actions/views to implement:

- **index** – a view with a heading (`<h1>`) “Ownership Records” and a table (`<table>`) with a labeled column for each attribute above and a row for each record stored in the database.
- **new/create** – a “New Record” (`<h1>`) form and a corresponding `create` action that redirects to the index page on success. There should be a create hyperlink on the index page above the table that takes a user to the “new” form.
- **delete** – a controller action that destroys a specified record. There should be a hyperlink in each row of the index table that, when clicked, destroys the record on that row.

Task 2

Create controller actions/views to manage this model class:

UserProfile
name : string birth_year : integer phone_number : string region : string

Controller actions/views to implement:

- **index** – a view with a heading (<h1>) “User Profiles” and a table (<table>) with a labeled column for each attribute above and a row for each profile stored in the database.
- **edit/update** – an “Edit User Profile” (<h1>) form and a corresponding `update` action that redirects to the index page on success. There should be an “edit” hyperlink in each row of the index table that, when clicked, takes the user to the edit form for the profile on that row.
- **delete** – a controller action that destroys a specified profile. There should be a hyperlink in each row of the index table that, when clicked, destroys the profile on that row.

Task 3

Create controller actions/views to manage this model class:

Review
title : string body : text score : decimal

Controller actions/views to implement:

- **index** – a view with a heading (<h1>) “Reviews” and a table (<table>) with a labeled column for each attribute above and a row for each record stored in the database.
- **new/create** – a “New Review” (<h1>) form and a corresponding `create` action that redirects to the index page on success. There should be a “new” hyperlink on the index page above the table that takes a user to the “new” form.
- **delete** – an controller action that destroys a specified review. There should be a hyperlink in each row of the index table that, when clicked, destroys the review on that row.

Task 4

Create controller actions/views to manage this model class:

Automobile
model : string model_year : integer kbb_url : string

Controller actions/views to implement:

- **index** – a view with a heading (<h1>) “Automobiles” and a table (<table>) with a labeled column for each attribute above and a row for each automobile stored in the database.
- **edit/update** – an “Edit Automobile” (<h1>) form and a corresponding `update` action that redirects to the index page on success. There should be an “edit” hyperlink in each row of the index table that, when clicked, takes the user to the edit form for the automobile on that row.
- **delete** – an controller action that destroys a specified automobile. There should be a hyperlink in each row of the index table that, when clicked, destroys the automobile on that row.

Task 5

Create controller actions/views to manage this model class:

Manufacturer
name : string phone_number : string email : string

Controller actions/views to implement:

- **index** – a view with a heading (<h1>) “Manufacturers” and a table (<table>) with a labeled column for each attribute above and a row for each manufacturer stored in the database.
- **new/create** – a “New Manufacturer” (<h1>) form and a corresponding `create` action that redirects to the index page on success. There should be a create hyperlink on the index page above the table that takes a user to the “new” form.
- **delete** – a controller action that destroys a specified manufacturer. There should be a hyperlink in each row of the index table that, when clicked, destroys the manufacturer on that row.