

GEDİZUNIVERSITY

Computer Engineering Department

COM401 (Fall-2014)
Software Engineering
Laboratory

HOMEWORK3 –

SAMPLE SOLUTION :



You are supposed to extend the myrottenpotatoes example we developed in the lab. The extension is adding a search *functionality by title* for the movies.

As logical, you should follow the steps below to add this functionality.

Step 1: Add a search division into the “index.html.erb” by implying _search.html.erb partial form is rendered in this division.

You can do this by adding the following lines.

```
<body>


Step 2: Create a new partial form titled “_search.html.erb” under “views/movies” folder so that we can see the search box on the “index.html.erb” file after rendering.



Add following lines to “_search.html.erb”



```
</br>
<%= form_tag(movies_path, :method => "get", id: "search-form") do %>
 <%= text_field_tag :search, params[:search], :id => 'search_field', :value => ''%>
 <%= submit_tag "Search", :name => nil %>
<% end %>
</br>
```



After adding this box, you should see such a screen on your homepage.



The screenshot shows a web browser window with the URL 127.0.0.1:3000/movies. The page has a yellow header bar with the word 'Movies'. Below it is a table with columns: Movie Title, Rating, Release Date, and More Info. The table contains several movie entries. At the bottom left, there is a link 'Add a new Movie'.



Movie Title	Rating	Release Date	More Info
Aladdin	G	1992-11-25 00:00:00 UTC	More about <a href="#">Aladdin</a>
When Harry Met Sally	R	1989-07-21 00:00:00 UTC	More about <a href="#">When Harry Met Sally</a>
The Help	PG-13	2011-08-10 00:00:00 UTC	More about <a href="#">The Help</a>
Raiders of the Lost Ark	PG	1981-06-12 00:00:00 UTC	More about <a href="#">Raiders of the Lost Ark</a>
Spiderwick	PG-13	2009-11-03 00:00:00 UTC	More about <a href="#">Spiderwick</a>
ar	G	2014-11-10 00:00:00 UTC	More about <a href="#">ar</a>



Sample Solution-PostLab3



Page 2


```

Step 3: Add a search function by movie title into movie model("models/movie.rb")

```
def self.search(query)
  # where(:title, query) -> This would return an exact match of the query
  where("title like ?", "%#{query}%")
end
```

After this addition, your movies model should look like below.

(Note that validates :title, :presence => true

line prevents a movie entry with empty title to the database.)

```
class Movie < ActiveRecord::Base
  validates :title, :presence => true
  has_many :reviews

  def self.search(query)
    # where(:title, query) -> This would return an exact match of the query
    where("title like ?", "%#{query}%")
  end
end
```

Step 4: We need to update our **movies controller** so that it can change by search results. It is enough to implement change index method, because we list all movies in the index method.

```
def index
  if params[:search]
    @movies = Movie.search(params[:search]).order("created_at DESC")
  else
    @movies = Movie.all
  end
end
```

These lines if there is a search parameter, Movie table searched for the parameter, then the results are listed in descending order. If there is no search parameter, in that case all movies are listed as usual.

You can find a sample source code for it on my website.

<http://arzumkaratas.weebly.com/sweng2014-2015.html>

"On my honor I have neither given nor received unauthorized information regarding this work, and I am unaware of any violation of the Honor Code by others."