AGGIUNGERE CONTROLLO PASSWORD

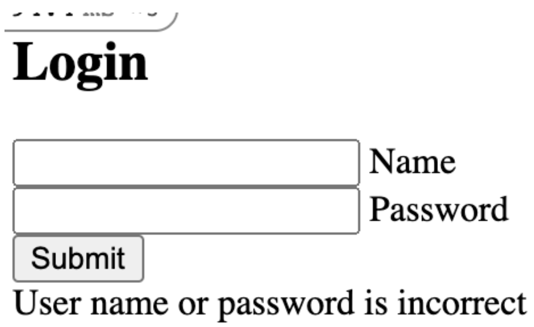
```
Moviegoer.create(:name=> 'test', :password=>'test')
```

CONTROLLO PASSWORD

```ruby
class SessionsController < ApplicationController
        def create
                name = params[:user][:name]
                @current_user = Moviegoer.where(:name=> name)
                if @current_user.length==0 or
                   @current_user[0][:password]!=password
                then
                        flash[:loginerror]=true
                        redirect_to login_path
                else
                        session[:user]=@current_user[0][:name]
                        redirect_to movies_path
                end
        end
end
```

FLASH

```
- if flash[:loginerror]
  = 'User name or password is incorrect'
```



User name or password is incorrect

VERIFICARE CHE TUTTE LE RICHIESTE PROVENGANO DA CLIENT AUTENTICATI…

```
class ApplicationController < ActionController::Base
    before_action :check_user_logged_in

    def check_user_logged_in
        if session[:user]==nil then
            redirect_to login_path
        end
    end
end
```

..TRANNE LA RICHIESTA DI LOGIN (SKIP ACTION)

```
class MoviegoersController < ApplicationController
   skip_before_action :check_user_logged_in
   def show
   end
end
```

.. E DEL CONTROLLO CREDENZIALI

```
class SessionsController < ApplicationController
        skip_before_action :check_user_logged_in
        def create
                name = params[:user][:name]
                password = params[:user][:password]
                @current_user = Moviegoer.where(:name=> name)
                if @current_user.length==0 or
                   @current_user[0][:password]!=password
                then
                        flash[:loginerror]=true
                        redirect_to login_path
                else
                        session[:user]= name
                        redirect_to movies_path
                end
        end
end
```

COME CANCELLARE UNA SESSIONE

Aggiungere route:

```
post '/logout' => 'sessions#destroy'
```

Aggiungere Pulsante in index.html.haml:

```
%br
%br
= form_tag logout_path, :method => :post do
  = submit_tag 'Logout'
```

Aggiungere nel controller Sessions:

```
def destroy
   session.delete(:user)
   redirect_to movies_path
end
```

GENERAZIONE REVIEW

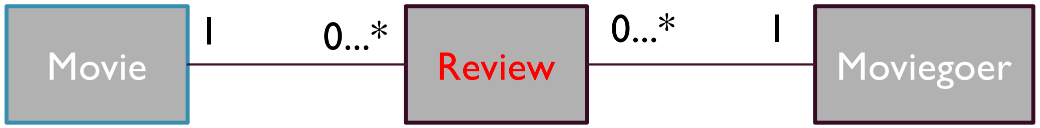
```
rails g migration create_reviews potatoes:integer comments:text moviegoer:references
movie:references
```

```
rails db:migrate
```

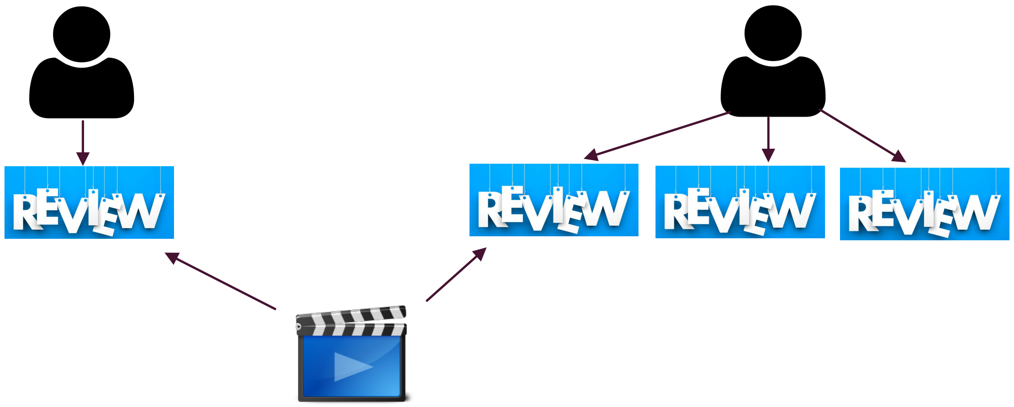
```
class CreateReviews < ActiveRecord::Migration[6.1]
  def change
    create_table :reviews do |t|
      t.integer :potatoes
      t.text :comments
      t.references :moviegoer
      t.references :movie
      t.timestamps
    end
  end
end
```

ASSOCIAZIONI

```
class Review<ActiveRecord::Base
    belongs_to :movie
    belongs_to :moviegoer
end
```



```
class Movie < ApplicationRecord
   has_many :reviews
end
```



```
class Moviegoer < ApplicationRecord
   has_many :reviews
end
```

ESEMPIO (DA RAILS CONSOLE)

```
Movie.create(:title => 'Inception')      Inception  = Movie.find(39)
```

Chiave primaria (trovata con Movie.all)

```
alice =  Moviegoer.create(:name => 'Alice', :password => 'alice')

bob = Moviegoer.create(:name => 'Bob', :password => 'bob')
```

Chiave primaria

```
Review.create(:potatoes => 5, :moviegoer_id => 5 , :movie_id => 39)

Review.create(:potatoes => 5, :moviegoer_id => 6 , :movie_id => 39)
```

REVIEW CONTROLLER

```
rails g controller Reviews new create destroy
```

MODIFICARE ROUTES (CONFIG/ROUTES.RB)

```
resources :movies do
  resource :reviews, only: [:new, create,:destroy]
end
```

```
rails routes -c Review

Prefix Verb    URI Pattern                                     Controller#Action
   reviews_create GET    /reviews/create(.:format)                 reviews#create
  reviews_destroy GET    /reviews/destroy(.:format)                reviews#destroy
new_movie_reviews GET    /movies/:movie_id/reviews/new(.:format) reviews#new
    movie_reviews DELETE /movies/:movie_id/reviews(.:format)     reviews#destroy
                  POST   /movies/:movie_id/reviews(.:format)     reviews#create
```

FORM DELLA REVIEW

./APP/VIEWS/REVIEWS/NEW.HTML.HAML

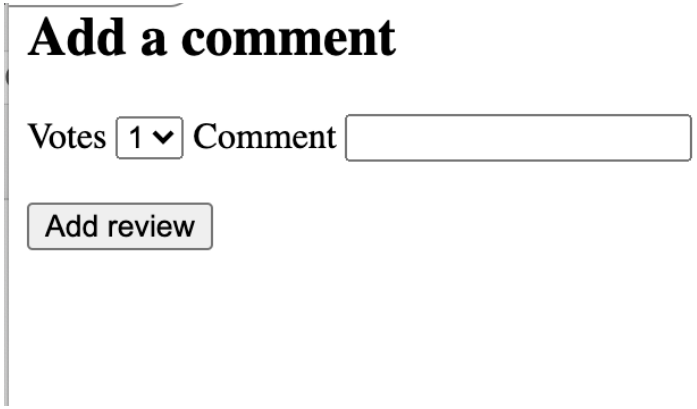
```
%h2 Add a comment

= form_tag movie_reviews_path, :method => :post do


  = label :review, :potatoes, 'Votes'
  = select :review, :potatoes, [1,2,3,4,5]


  = label :review, :comments, 'Comment'
  = text_field :review, :comments


  <br/><br/>
  = submit_tag 'Add review'
```



Add a comment

Votes 1 Comment

AGGIUNGER LINK NELLA VIEW SHOW.HTML.HAML

```
%br
#{link_to 'Add comment',
new_movie_reviews_path(@movie)}
```

CREARE UN COMMENTO

```
def create
        id_movie = params[:movie_id]
         user_id = session[:user_id]
        @movie = Movie.find(id_movie)
        @review = @movie.reviews.create!(:comments =>
params[:review][:comments],:moviegoer_id=> user_id)
        redirect_to movies_path
  end
```