Moocme

By: Dor Rondel

Project Description:

For a myriad of reasons, many individuals curious about the abundance of knowledge universities have to offer, cannot allow themselves to claim any of it. A relatively recent phenomenon, in 2011, Stanford published three of their courses online for free, and set the wave for a new means of sharing higher education. Today there are hundreds of university courses and thousands of independently taught courses, or MOOCs, offered on a multitude of platforms, such as Udacity, Coursera, Udemy and EDx amongst others. This is where MOOCme comes in. In its essence, MOOCme is a website hosting a MOOC search engine. This search engine will supply MOOC students all over the world the leisure of searching for desired courses, on multiple platforms, from a single site.

System Requirements:

Functional Requirements:

Identifier	Priority	Requirement
REQ1	5	Site must be able to retrieve all MOOCs from at least the top MOOC platforms mentioned in the project description.
REQ2	5	Site must return the current MOOCs corresponding to the search terms.
REQ3	3	Site must be able to filter MOOCs from the different platforms.
REQ4	3	Site must be able to filter free and paid content.
REQ5	4	MOOCs must link to their source.
REQ6	3	Site must have user profiles.

REQ7	3	Site must be able to hold onto courses student chooses to save.
REQ8	2	Site must recommend a relevant course to the user.
REQ9	3	Site must have a functional authentication mechanism.
REQ10	5	Site must be accessible to everyone and remotely hostable.

Non-functional Requirements:

Identifier	Priority	Requirement
nREQ1	3	Site must be have a nice UI/UX.
nREQ2	2	Site must be hostable on some IaaS.
nREQ3	2	Site will be responsive.
nREQ4	4	Code will be stored on github.
nREQ5	4	Searches won't take longer than 20 seconds.
nREQ6	3	Site can handle heavy user load.

MOOCme API:

Title: Home visit
URL: /
Method: GET
Data Params: [Boolean] authenticated
Returns: HTML document

Title: Signout
URL: /
Method: POST
Data Params: [Boolean] authenticated
Returns: HTML document || 403

Title: Search submission
URL: /search?=<encoded-string>
Method: POST
Data Params: [String] search, [Boolean] Authenticated, [Boolean] free
Returns: HTML document

Title: Signin request
URL: /signin
Method: GET
Data Params: [String] search, [Boolean] Authenticated,
Returns: HTML document || 307

Title: Signin submission
URL: /signin
Method: POST
Data Params: [String] username, [String] password
Returns: JSON: {
        Authenticated: [BOOL] authenticated
}

Title: Signup request
URL: /signup
Method: GET
Data Params: [String] search, [Boolean] Authenticated,
Returns: HTML document || 307

Title: Signup submission
URL: /signup

Method: POST
Data Params: [String] username, [String] password
Returns: JSON: {

      Authenticated: [BOOL] authenticated

}

Title: Profile view
URL: /profile/:id
Method: GET
Data Params: [Boolean] Authenticated
Returns: HTML document || 403

Title: Add course
URL: /profile/:id
Method: POST
Data Params: [Boolean] Authenticated
Returns: 201 || 403

Title: Remove course
URL: /profile/:id
Method: DELETE
Data Params: [Boolean] Authenticated
Returns: 201 || 403

Title: About page
URL: /about
Method: GET
Returns: HTML document

Sequence Diagram:

CLIENT	MIDDLEWARE	SERVER

A) Home Requested
B) Search Submitted
C) Course Added
D) Course Deleted
E) User Signin
F) User Signup
G) User Sign out

User Request Sent to be Processed

User Sees Response from the Server

Server Sends Data to User