WEBAPDE Machine Project AY1718T3

Machine Project

Your task is to create a web application of your choice from the options listed below.

NOTES

- The group must be composed of 2-3 students.
- There are 3 MP options listed below. A group must choose one for phase 1. A group may switch their option for the next 2 phases. Completing all 3l will not incur any additional marks.
- If there are any questions, or the specifications seem vague, approach your teacher immediately for clarifications.

PHASE 1

- The application will be built using HTML, CSS, and JS.
- The group may use their preferred JavaScript libraries.
- The group may use their preferred interface libraries like Bootstrap and Meteor.
- All views all users will see in the final version must be visible. All views must be navigable from the first page.
- Data can be hardcoded for now.
- Prepare at least 5 dummy data for each collection (ie life-like data, avoid lorem ipsum) to fill in the website.
- Include a site-map of the website.

PHASE 2

- The project option for phase 2 must match the project option for phase 3.
- Architecture need not implement MVC yet, but highly encouraged. For encouragement, MP3 is required to be MVC.

Model

- Database design must be complete and submitted.
- Model modules of the project must be connected to the database. All necessary CRUD operations for all collections are expected to be complete.
- All passwords should be hashed and database executions must account for injection attacks.

View

- All views must be visible and can be navigated to from the first page.
- All forms must use the appropriate HTTP methods (ie Login forms use POST, search queries use GET).

Controller

- The back-end is expected to be NodeJS. Database dialect is encouraged to be MongoDB (Mongoose as an ORM is accepted). The use of a different server side scripts and database is not allowed.

PHASE 3

- The complete program must be working with all the features running.
- The system must be deployed. Deploying to Heroku is recommended.
- Project must follow the model-view-controller architecture.
- Content seen/manipulated in the views must be reflected in the database.
- There should at least be 20 dummy data and 4 user accounts.

- Sessions must be properly implemented.
- The system must check for cross-site scripting.
- All NPM packages used for the project as well as 3rd party libraries must be stated and indicated in the "About" page.

SUBMISSION

Phase	Submission
Phase 1	13 July 2018 (M) Canvas
Phase 2	7 August 2019 (M) Canvas
Phase 2	14th week (TBD based on student availability) G301

OPTION 1: MEME SHARING SITE FEATURES

View all public photos

Upon visiting the web page, an unregistered visitor may see the 15-20 most recently uploaded public memes. The user can see the next set of recently uploaded memes, it is up to the group whether to implement this in the same (auto-load) or another page. Clicking the thumbnail will let the user view the meme in full-view (*see:* **view a meme**).

Apart from the public memes, the user will also see memes shared with them in a separate section in the home page. (*see*: **share a photo**)

View a user profile

Each user has their own page which shows their profile publicly. On the same page, a visitor may see the user's username and their short description. They may also see a portion of the user's latest memes. The visitor may opt to see the rest of the memes. It is up to the group if this going to be loaded on the same page or a different one.

Register

A visitor must register if they want to upload. Here, a visitor must enter their username, their password, and a short description (can be left empty).

Log in

After registering properly, a visitor may log-in. Upon logging in, the user can start uploading their memes. The user is given the option to be "remembered" by the website. When the user chooses this option, every log in and visit to the website will extend their "remember" period by 3 weeks.

Note The user needs to be logged in before seeing any meme they own or given access to. If the user is logged in and views a meme, the URL leading to that meme should throw a restricted error when the user is not logged in.

Log out

The user may log out from their account. This should cut short the "remember" period if it exists, and clears any session-related data.

Upload a meme

A user may upload a meme (JPG, PNG, TIFF) from their local drive. They must give a dank title for the meme, and tags about the photo (*see*: **tag a meme**). The user must also choose whether the meme is public or private. The tags must be structured properly as this will be used later for searching.

Note Make sure that the uploaded file is a meme of the allowed formats.

Share a meme

A user may give another user privilege to view their meme by adding the other user's username to the "allowed" list of the photo. This option is only available if the photo is private.

wView a meme

A visitor/user can see a meme's details as long as it is publicly shared, or shared to them. The meme will be shown in full resolution, with the title, description, uploader, and the tags associated to it. At this stage, the user who owns the meme may also append more tags (*see*: **tag a meme**).

Tag a meme

A user who owns the meme may add tags to a meme when uploading or viewing the meme.

Search a meme by tags

A visitor/user can search for memes using tags. By entering a tag, all memes with that tag will appear as results.

General

Good user experience. Visitor can easily navigate without help, all information are easy to access.

Good visual design. Design suits the theme of the application, and is cohesive and consistent across the whole application.

OPTION 2: FORUM
FEATURES

View all posts
Upon visiting the web page, an unregistered visitor may see the 15-20 most recently* uploaded post titles and a post description snippet. The user can see the next set of uploaded posts, it is up to the group whether to implement this in the same (auto-load) or another page. Clicking the post title will let the user view the post and the comments (*see:* **view a post**).

* The user may also see the most popular post based on a calculated ranking.

View a user profile
Each user has their own page which shows their profile publicly. On the same page, a visitor may see the user's username, avatar and short description. They may also see a portion of the user's latest posts and comments. The visitor may opt to see the rest of the posts and comments of the user.

Register
A visitor must register if they want to post or comment. Here, a visitor must enter their username, their password, an avatar, and a short description (can be left empty).

Log in
After registering properly, a visitor may log-in. Upon logging in, the user can start posting and commenting. The user is given the option to be "remembered" by the website. When the user chooses this option, every log in and visit to the website will extend their "remember" period by 3 weeks.

Log out
The user may log out from their account. This should cut short the "remember" period if it exists, and clears any session-related data.

Post
A user may make a text post. They must give a title for the post, and the body of the post. Additional points will be given for allowing markup (full) without the risk of cross site-scripting.

View a post
A user may view any post they have a link to. This will load the title, the body of the post, and the comments of the post.

Comment
A user can comment on any post, including their own. They may also reply to another user's comments, and the comments can nest indefinitely.

Edit a post
The owner of the post may edit their posts at any point.

Edit a comment
The owner of the comment may edit their comments at any point.

Delete a post
The owner of the post may delete their post.

Delete a comment
The owner of the comment may delete their comment.

Upvote
A user can upvote a post or a comment (including their own) once.

Downvote
A user can downvote a post or a comment (including their own) once.

Search a post
A visitor/user can search for posts by similarities in the title or the post body. By entering a search phrase/word, all posts containing it will appear as results.

General
Good user experience. Visitor can easily navigate without help, all information are easy to access.

Good visual design. Design suits the theme of the application, and is cohesive and consistent across the whole application.

OPTION 3: Lab reservation system

Reservation system for 2 computer laboratories (20 machines each). Lab slots are in intervals of 30 minutes.

FEATURES

View lab availabilities

A user may choose a computer lab and see the current available seats of the chosen lab. The user may opt to see the availability of the lab at other times for the next 7 days. A user must be a student user to reserve a seat.

Note Users may not see who reserved a seat. The availabilities must also update periodically so that the user does not need to refresh every time.

Register

A visitor must register if they want to reserve a lab slot. Here, a visitor must enter their DLSU email, and a password. There are should be 2 kinds of accounts, the student who can reserve lab slots, and the lab technician that can block a time slot for walk-in students.

Note Email must be confirmed before the student can reserve.

Log in

After registering properly, a visitor may log-in. Upon logging in, the option to reserve a slot will be open. The user is given the option to be “remembered” by the website. When the user chooses this option, every log in and visit to the website will extend their “remember” period by 3 weeks.

Reserve

Students can reserve slots that have not yet been taken. Lab slots are in intervals of 30 minutes. The student can reserve more than one slot if this is not enough, and all the lab slots will be made under one reservation. A student may not reserve a previously reserved slot.

Note The reservation must be composed of consecutive lab slots (no skipping in the middle). The reservation cannot be made for a past date.

Reserve for a student

Lab technician can make a reservation for walk-in students. The reservation can be a string

Note The reservation must be composed of consecutive lab slots (no skipping in the middle). The reservation cannot be made for a past date.

Remove reservation

Lab technicians have the ability to remove reservations of students who do not show up within 10 minutes of the reservation. This facility is only available 10 minutes of the actual reservation time. This will cancel the whole reservation.

See reservations

A user may check their reservations. They can see details such as the seat number, the laboratory, the date and time of request, and the date and time of reservation.

Search for free slots

Users can view all the available slots given a provided date and time, and the lab.

Edit a reservation

A student/lab technician can edit reservations they previously made.