360 Final Project

Declared requirments

System Requirements:

```
MAJOR (Foundational):
    - Hand-styled layout with contextual menus. (i.e. when a user has logged on to the site, menus reflect the change).
        Menus are responsive to user changes, eg they can seee their own comments after posting

    - 2 or 3-column layout using appropriate design principles. (i.e. highlighting nav links when hovered over, etc) (Everyone)
        We used a two colum layout for our desktop site and a one column for our mobile, just to improve user experience

    - Form validation with JavaScript. (Everyone)
        All forms are properly validated

    - Server-side scripting with PHP. (Everyone)
        all user input is sanitized before usage

    - Data storage in MySQL. (Everyone)
        All user data apart from posts/profile pictures are stored in the database directly

    - Appropriate security for data. (Everyone)
        all user inputs are sanitized and passwords are properly hashed before being stored

    - Site must maintain state. (user state being logged on, etc) (Everyone)
        Site utilizes sessions to make sure that the user isnt logged out until they leave for an extended period.

    - Responsive design philosophy. (minimum requirements for different non-mobile display sizes) (Adam)
        relative units are used where appropriate and the website is very capable of resizing as well as becoming a mobile site

    - AJAX (or similar) utilization for asynchronous updates. (meaning that if a discussion thread is updated, another user who is viewing the same thread will not have to refresh the page to see the update) (Adam)
        The page progressively fetches posts and comments according to the filter being used and are fetched in groups of 25 as the user scrolls so that it doesnt flood the site with every post in the database, and inserts
```

```
new posts that you had missed due to the sort so that new posts are always shown to the user

    - User images (thumbnail) and profile stored in a database. (Tanner)
        I opted to store the image files on the server, and the uniquely generated file paths to the database for access later. I also experimented with storing them as BLOBs, but I preferred the former.

    - Simple discussion (topics) grouping and display. (Adam)
        Comments are isolated between posts and posts are easily grouped within their communities/forums

    - Navigation breadcrumb strategy. (i.e. users can determine where they are in threads) (Joy)
        Path between windows is tracked using server sessions

    - Error handling. (bad navigation) (Everyone)
        All of our php pages are protected fully and redirect to appropriate pages, eg trying to view forums while not logged in

    - Allow seperate forums for different clubs. (Joy)


MINOR (Additional):
    - Search and analysis for topics/items. (Adam)
        The search bar finds all near matches in a title for a given string using "LIKE" functionality

    - Hot threads/hot item tracking. (Adam)
        The user can filter by promos using a request to the server for new posts that are loaded with AJAX

    - Activity by date. (Adam)
        The user can filter by date Ascending or Descending

    - Collapsible items/treads without page reloading. (Adam)
        Comment sections are collapsable by the user using Javascript to begin/stop requesting comments from a specific post

    - Alerts on page changes. (Joy)
        Page will give alerts for new, unviewed posts by flashing the home button

    - Styling flourishes. (Adam)
        While The entire team did help with Styling It was up to me to refine it, so that it was consistent and visual bugs were close to absent

    - Responsive layout for mobile. (Adam)
        By Using media queries and scripts to make our menus into overlays I was able to create a well built mobile site!
    - Tracking comment history from a user’s perspective. (Tanner)
        Users can view their own comments in the dashboard intermixed
between posts

    - Accessibility. (Alt text for images, screen reader support?) (Adam)
        We gave users the ability to add titles to describe their posts
and labeled everything with proper landmark roles to support the usage of
screen readers
```

User Requirements:

Unregistered Users:

```

MAJOR:
    - Allow users to register with an email and password (Tanner)
        Users can register on the register page, all fields are validated
with javascript.

    - Allow users to view content without an account (Everyone)
        Users can view popular and posts/comments without an account

MINOR:
    - Allow Forum managers to "Pin" posts so that they are seen first when
the forum is viewed.
        Forum Managers can pin posts in the community page with a php
script that marks them

    - AJAX posts marked as threads, so users may interact with each other
live.(Adam)
        All comment sections are live updated with the most recent
comments using polling and progressively loading comments as the user
scrolls

    - Collapsable threads. (Adam)
        Comments sections, and option menus in mobile sites are
collapsable. The User can clear the comments by clicking the close button
using javascript to fetch the comments of the clicked comments button from
a php script
```

Registered Users:

```

MAJOR:
    - Show User login state through manipulation of UI elements (Adam)
        Login Button is replaced by the username of the logged in user by
using php to print it when the page is refreshed

    - Allow Users to log out (Tanner)
```

```
        Users can log out from the header using a simple PHP script that clears the server session

    - Allow Users to delete their accounts (Tanner)
        Users can delete their account from the Account page after using their password to reverify themselves

    - Allow users to like content. (Adam and Tanner)
            Users can "Promo" content to give it a boost on popular sorting by the page making a js fetch request to the server and waiting for confirmation

    - Allow Users to sort posts by popular, user interaction, and date (Adam)
        Users Can sort by the amount of promos(Likes), oldest or newest first, or lack of promos

    - Allow Users to comment on posts. (Adam)
        Users can comment on any posts by Js making a fetch request with the contents of the message to the server

MINOR:
    - Allow Users to delete comments and posts (Tanner)
        Users can delete their own comments from anywhere on the site, and delete their posts from their account dashboard. Site Admins can delete all comments.
```

Administrators:

```

MAJOR:
    - SAdmin should be able to delete users from the site. (Tanner)
        Site admins possess the ability to delete all posts, users, and comments on the wbesite.

    - FAdmin should be able to ban/kick users from forums.
        Forum admins possess the ability to delete all members in the forum.

    - FAdmin should be able to view/remove posts on forums.
        Forum admins can view as well as delete the post in the forum.
MINOR:
    - FAdmin should be able to invite people to moderate.
        Forum admins can view the member list and assign them to moderator.

    - FAdmin should be able to remove moderators that are newer than them.
        Forum admins can unassign users from the moderator role.


    - FAdmin should be able to pin posts.
        Forum admins can pin the posts to be seen first.
```

Additional functionality

Search and analysis for topics/items

```
Posts are searchable by using the titles, and can be filtered while
searched
```

Hot threads/hot item tracking

```
Items can be searched based on the number of promos in a way to find
hot/popular items
```

Visual display of updates, etc (site usage charts, etc)

```
The site Flashes the home button to alert the user to new posts available
in the feed
```

Activity by date

```
the site sorts comments/posts by date automatically and the
```

Tracking (including utilizing tracking API or your own with visualization tools)

```
We did not utilize an external tracking API, but we did implement member
tracking for communities in order to best understand usage
```

Collapsible items/treads without page reloading

```
comment sections are collapsed back into the post after clicking the close
button
```

Alerts on page changes

```
The server alerts the client on changes in the feed by flashing the home
button
```

Admin view reports on usage (with filtering)

```
Admin can view and delete any activity on the site, so they can use the
forums page to sort and view usage, there are no specific reports as we
did not know what we would be reporting on
```

Styling flourishes

```
Styling is insanely cool and awesome, I mean look at the mobile site.
Clearly the best
```

Responsive layout for mobile

```
Our site is completely compatible with mobile. having its own dedicated
site design but the responsiveness doesnt stop there. The site can easily
be resized to fit any screen size or browser size without a hitch
```

Tracking comment history from a user’s perspective

```
Users can view any of their posts/comments from their profile page
```

Accessibility

```
The site is properly Setup for Landmark roles for screen readers, and the
usage of user titles helps to describe content for said posts, allowing
screen readers to scrape posts semi consistently. there could be some
improvements as we did not have the full time to experiment with screen
readers but we gave it a good effort I think. We also used a bright and
highly contrasting color pallete to improve usage for our color blind
users
```

Known Limitations

```
Currently, there is no way to create admin accounts witout manually
cahnging their privelege value to "2" in the database.
Only Admin accounts can search for users, regular accounts cannot find
other users.
Joining a community doesn't update the user's homepage feed.
Upvoting your own posts from the account dashboard is uavailable.
```

Extra Marks and special Features

```
Our Team believes we deserve the highest possible grade, as not only are
all of our sites abilities fully functional, but we also took great
consideration for scalability and efficiency. Our site doesnt store any
posts or images in the database, allowing us to easily scale and implement
new features for our users to edit their own posts and such. The AJAX Adam
implemented also loads posts progressivly, stopping the server from
overserving a client, and stopping a clients machine from being flooded as
the posts database and comments database grows.
```

AJAX Post and comment fetching

```
Our AJAX Post fetching Uses a last fetched time and a scroll detector to
load new posts as the user scrolls, allowing them to keep viewing posts
without being interupted
The last fetch time is used to get new comments/posts without refreshing
the page, or placing them near the top of the page away from the users
view.
This serves to minimize the client and Server resources for loading posts
and comments, as it only loads enough for the user to view at one time.
This also keeps all feeds 'Live' in that the user never needs to refresh
to see the newest posts
```

Mobile site

```
Our Mobile Site is awesome, looking nearly like a full webapp, with pop up
menus and proper styling
What more needs to be said?
```