

Project - Phase 2

Functionality

For phase 2 you will build the core functionality of your app and integrate a database into your architecture.

GitHub Repository

Commit and push all contributions to your existing repository. Update your code, readme, and Open-Source Reports as needed.

Functionality

The primary programming task for this phase is to build all the functionality of your app that you can build without implementing user accounts (Adding user accounts is phase 3). As you design your app, it is highly recommended that you consider how user accounts will be added later. Do not design your app in a way that is not compatible with user accounts since this will create much more work for your team during phase 3.

For each of the [3 of the] 5 project criteria, you must have the following by the end of phase 2.

- User Accounts
 - Your entire app should operate under the assumption that there is only a single user. You don't need any authentication and anyone visiting the page can use every feature
 - Note: It is fine if you do add some sort of user accounts to get ahead for phase 3. For example, allow users to enter a username without authenticating them and have your backend treat them as separate users
- Live sharing of multimedia content between users
 - Content can be uploaded, saved on the server, and downloaded. The content must persist through a server restart
 - When a user uploads content it must efficiently appear on other open tabs/windows without requiring a refresh. This must be done with either WebSockets (Recommended) or Long-Polling. Use of Polling will result in lost points
 - You must store some record of each submission in your database. It is recommended to store any meta-information you need (content id, etc) along with the filename where the content is stored on disk. You do not have to store the content itself in your database
- Users can "vote" and comment on user-submitted content
 - Voting and commenting features must be fully functional
 - When a user votes/comments it must efficiently appear on other open tabs/windows without requiring a refresh. This must be done with either

WebSockets (Recommended) or Long-Polling. Use of Polling will result in lost points

- Votes and comments must be stored in your database and persist after a server restart
- Users can friend or follow other users
 - Not required for phase 2
- Users can send direct messages to other users
 - Not required for phase 2

Security

Your site must be secure! If a vulnerability is found, the consequences will be decided based on the severity of the vulnerability. Severe vulnerabilities may result in a 0 for this phase of the project.

We will explicitly check for the following during grading:

- HTML/JS Injection (This is considered a severe vulnerability!)
- SQL Injection (Use prepared statements)

Note: For phase 3 this list will be much longer since you will be handling user data including their passwords.

Docker-Compose

Your app must be setup to deploy with docker-compose. In the root directory of your repository include a docker-compose.yml file with everything needed to run your app using docker-compose. At a minimum this must include creating a container for your database and another container for your app that will communicate with the database container.

- You may choose the local port for your app

When testing or deploying your app the procedure will be:

1. Clone your repository
2. cd into to the directory of the repository
3. Build the docker image (docker-compose build)
4. Create and run a docker containers (docker-compose up --detach)
5. Open a browser and navigate to `http://localhost:<local_port>` (We'll read your docker-compose.yml to find your local port)

Video

Create a video demo of all the functionality of your site. This video has the same requirements as the video for phase 1, though be mindful that you are showing a more complex piece of software and you may need more explanation to accurately demo your app (Ex. If you didn't have voice over for phase 1 you may want to add it for phase 2)..

Submission

Fill out [this form](#) (You only need 1 submission per team) with the url for your YouTube video.