

Knowledge Base - 4711

Document Version 1

- Gibeom (Curtis) Kim - **Profiles** curtisk808@gmail.com
- Alex Hosseini - **Database and Authentication** - a.hosseinyy@gmail.com
- Youngbin (Johnny) Kim - **Posts** - youngbeanies@gmail.com
- Phillips Halim - **Messaging** - phillipshalim@gmail.com
- Yue (Tony) Wing Lau - **Integration** - tonyywlau@gmail.com

We will be communicating as a team through a WhatsApp group that we have created. Most communication will be conducted in this group.

Project Description

A social media sharing site of knowledge base articles. Users will be able to sign up and begin posting questions/issues. Other users can comment on these posts. Users will also be able to send messages and 'like' other's profiles. It looks like LinkedIn mixed with Stack Overflow.

Technical assumptions:

We will be creating this social media website using MVC architecture. Our web application will utilize technologies such as HTML, CSS, Javascript, Node.js. Regarding storing the user data, we are planning to store the data in an SQL database hosted on Heroku. The server that hosts the website is being planned to be hosted on Heroku as well. We plan on using Github as a team for version control. We plan on creating this website with all the functionality described in the document.

Non-Technical Assumptions:

We plan on creating the look and feel of this website as similar to the document described as possible. We plan on meeting weekly to work on and discuss the project. We will also communicate in our WhatsApp group afterward when working on the project.

Routes

/signup	ProfileController -> createProfile ProfileController -> Login
/home	PostController -> getAllTopics
/user/profile/1	ProfileController -> getProfile
/user/message/1	MessageController -> getMessages
/topic?=id	PostController -> getTopic

Controllers

ProfileController <ul style="list-style-type: none">• createProfile• getProfile• updateProfile• deleteProfile• Logout• Login	For adding, updating and deleting for new users and login and logout for users. POST Request -> Store data into DB GET Request -> Fetch data from DB POST Request -> Update data POST Request -> Delete data Log out POST Request -> Check database -> Login
MessageController <ul style="list-style-type: none">• getMessages• sendMessage	For loading, getting, sending messages with others. Pull from the messages from the messages database. Send the message, updating the message database with the new message.
PostController <ul style="list-style-type: none">• getAllTopics• getTopic• getReplies• postTopic• postReply	For getting, adding post data messages from DB GET Request -> Fetch data from DB GET Request -> Fetch data from DB GET Request -> Fetch data from DB POST Request -> Store data into DB POST Request -> Store data into DB
searchController <ul style="list-style-type: none">• searchPosts	For searching and getting post data from DB GET Request -> Fetch data from DB

Views

login	Shows a form <ul style="list-style-type: none">• Email• Password• Login button
signup	Shows a form <ul style="list-style-type: none">• Firstname• Lastname• Password• Confirm password• Signup Button
signup_details	Shows a form <ul style="list-style-type: none">• Img url• Description• Date of birth• Country• Complete Registration button
home	Shows most recent posts, profile, and post message form
profile	Shows info related to user <ul style="list-style-type: none">• Posts• Replies• Likes• Picture• Name• description
post_topic	Shows Form <ul style="list-style-type: none">• Subject• Description• Topic (drop down)• Go button Shows Users profile on the left Shows latest discussions under the form

Models

User	Id Name Image_url Country Dob description	Number (Primary Key) String String String Date String
Post	Id Title Topic Description Author_id	Number (Primary Key) String String String Number (Foreign Key) - User table
Reply	Id Post_id Author_id	Number (Primary Key) Number (Foreign Key) - Posts table Number (Foreign Key) - User table
Message	Id Subject Content Sender_id Recipient_id	Number (Primary Key) String String Number (Foreign Key) - User table Number (Foreign Key) - User table

High Level Schedule / Risks / Testing

	Week 1 (Nov 13)	Week 2 (Nov 20)
Goal	<ul style="list-style-type: none">- Outline the project structure- Complete Initial Report- Create Git repo and assign tasks	<ul style="list-style-type: none">- Set up Models, Views, and Controllers- Add routing- Connect Database- Add Posts/Questions
Risks	Communication troubles Permission issues	<ul style="list-style-type: none">- Incorrect Routing- Persistence
Testing	NA	DB Access Test all possible routes, including

	Week 3 (Nov 27)	Week 4 (Dec 4)
Goal	<ul style="list-style-type: none">-Add replying to other posts-Add liking profiles-Add message sending	<ul style="list-style-type: none">-Complete the application-Test
Potential Risks	Integration Ignoring coding practices	Integration Minimize code redundant
Testing	Request, Response	Overall Flow