

## Design Report – Project AskMentor

AskMentor is online platform where user can post Computer science related questions, as well as provide answer, connect with other users through direct messaging, search post/channels/user and have personalized profile.

This is the containerized application with following packages/ tools used for development.

Packages/ tools	Purpose
Docker	Application containerization
Mysql2	Facilitates database management
Express	Framework for handling backend functionality
Cors	Enables Cross-Origin information sharing
Body-parser	Parses incoming request bodies
Multer	Handles file uploads
React-bootstrap	Provides react configured UI components for design
axios	Handles Promises-based HTTP requests
react-emoji-mart	Includes designed emojis
react-router-dom	Handles routing between different pages
react-textarea-autosize	Provides resizable text area

**Docker architecture** - I have created 3 containers for developing my applications. Following table will provide information about application containerization.

Container	Images	Purpose	Ports
Database	mysql:8.0-debian	MySQL database service	3306 :3306
node	Node :14	Backend application service	4000: 4000
react	Node: 14	Frontend application service	3000: 3000

**Backend architecture** - I have created “*postForum*” database for this application. This database contains following tables

Database Table	Purpose	Columns
userTable	Store information about users	username, email, password, name, profession, skills, avatar, totalPosts, connections,expertise, datetime
mediaTable	Stores details of social media accounts linked with user’s profile	userId, type, link, image(like social media platform logo)
channelTable	Store information of channels	name, totalPosts
postTable	Stores all posts and nested replies	userId, channelId, replyTo, topic, data, level (to recognize hierarchy) , likes, dislikes
fileTable	Store file details associated with post or replies or messages	postId, messageId, fileName, fileType, file (files contents)
messageTable	Stores messages between 2 users	senderId, receiverId, message, datetime

**Frontend Architecture** – For the frontend I have following components/ pages included. The application is available on <http://localhost:3000>

Components / pages	Path	Purpose
Homepage	/	Landing page that offers brief description about website
Homepage navbar	-	Contains website logo and sign in/login button
Channels page	/channels	Displays all channels, option to create new channel, all posts, list of connected users and brief profile details of current user
Messages page	/messages	Displays all connected users, messages between selected user and current user
Profile page	/profile	Display details of current user, option to edit profile, list of suggested channels and users
Navbar for all pages excluding homepage	-	Contains website name and logo, logout button and option to navigate between different pages

#### General functionality -

1. The landing page on <http://localhost:3000> provide brief description about the website including its features. User must sign in or log in to use the website. Landing page has two button one on navbar and other at foot bar to open the sign in/login modal, where user can input his/her information and log in/ sign in.
2. Once user is authorized to use website, it will redirect it to `/channels` page. On the left side bar user can see all the channels as well as button to create new channel. On the right-side bar, user can see brief description about his/her profile as well as list of connected users. Each item in the list has 'view profile' link which open offCanvas with selected user's profile and `message` link which will redirect to `/message` page. In the middle bar, at the top, there is button to let user create new post. User can include emojis and files as well in the post. The created post will be display below in middle bar including all other posts and their nested replies.
3. Each post has options to like, dislike and reply. With reply you can also add emojis and files.
4. The navbar contains search option which opens search modal, where user can search users/channels and posts. After clicking on searched item, it will navigate user to proper place.
5. There is one account hardcoded as a admin. When admin login into website, it can delete the posts / replies, remove users or channels.

#### Extra Features -

1. On `/profile` page, on the left side bar, user can see suggestions about active user and active channels. As well as this page has edit profile option where user can edit his/her profile including avatar, name, username, skills, professor and linked social media account.
2. Each item in list of users has `message` link. When clicking on that link user gets redirected to `/messages` page with selected user's account selected. User can send message to selected user. On the left side bar user can see list of all connected users and on right side bar user can see selected user's profile.