Project Plan

Task:

Build a complete full-stack application

Period:

11.08.2020 - 21.08.2020

Team:

Ingo Fischer & Felix Wurst

Description:

We have decided to develop a chat app. With this app we want to enable users to communicate with each other in real time. Besides sending text messages there are other possibilities like sending pictures and setting emojis. There is also a choice of different chat rooms. Beyond that, further rooms can be created. Our goal is to provide an intuitive user interface to delight the user with an easy to handle and fast communication channel.

Details:

This chat app is a realtime broadcast with WebSockets. It is a protocol that allows a bilateral synchronous exchange between a client and a server. We are using Node.js for webserver, Express for the gateway to one of the single chats. The chat is running with Socket.io. It is a library based on the WebSockets protocol to make the use of it easier. For the connection to a database mysql was used as the management system and phpMyAdmin as its administration tool. The application was built with Vanilla JS.

Steps:

1. find a project
2. integrate chat template
3. create a registration and login form
4. set up database design
5. combine database with webserver
6. build user handling in express
7. build room handling in express
8. build message handling in socket.io
9. build file handling in socket.io
10. create message history reload
11. integrate emoji selector
12. provide leave and logout options
13. improve frontend design
14. add validation for messages
15. provide chatroom creation
16. add validation for chatroom creation
17. handle events in socket.io with database
18. fix bugs
19. hosting
20. presentation

Schedule:

We have 14 days to realize our project (including two weekends). On Day 14 is the presentation.

Day 1:
- find a project, decide to make a chat-application

Day 2:
- begin to create an user-interface (registration)

Day 3:
- finish to create an user-interface (login, room-selection)
- begin to create tables for database (users)
- combine database with webserver
- start to build user handling in express (register user)

Day 4:
- finish to build user handling in express (login user)
- finish to create tables for database (rooms, messages, user_room)
- build room handling in express (get rooms)

Day 5:
- build message handling in socket.io (insert message)
- create message history reload
- integrate emoji selector

Day 6:
- provide leave and logout options
- improve frontend design (design for mobile, add buttons)

Day 7:
- start to build file handling in socket.io (send images)
- start to provide chatroom creation

Day 8:
- build file handling in socket.io (send images)
- finish to provide chatroom creation
- start to handle events in socket.io with database

Day 9:
- finish to build file handling in socket.io (send images)
- handle events in socket.io with database

Day 10:
- add validation for messages (use html-entities)

Day 11:
- fix problems (multiplied images, button-positions, converting from blob to string, limit image-size)
- add renaming of uploaded image-files
- finish to handle events in socket.io with database
- hosting

Day 12:
- create readme & project plan
- add validation for chatroom creation

Day 13:
- last fixes
- plan the presentation

Day 14:
- presentation