CLOUD COMPUTING

A simple chat Server

User Registration, Client side

- On Path „/“ the user gets an html file where he can decide if he wants to Register or Log In
- On „/register“ the Client has to choose a Username and a Password
- By submitting the formular, there is an Event Listener wich calls the function that fetches to the Server
- The function takes the Username and the Password from the Formular
- By using fetch, there is a Post request to the Server to path „/register“. In the Body of that request is the Username and the Password in a JSON File
- After the fetch Method, the function will ask for Status Codes so the Client knows if something went wrong
- On status code 200, the user gets redirect to the Chatroom on path „/chatroom“

User Registration, Server side

- Sending File register.html on path „/register“
- Taking the Username and the Password from clients request-body
- Security Checks from Clients Payload
- Hashing Password if everything is fine and store the Password and Username to Database
- Sending Status Codes to Client
- If everything went fine, the server will sent the file index.html

https-Conection

- Using the „https“ Component to use tls
- Created the Certificate and the public Key by using the openssl Command Line Tool
- To load the Certificate and the Key into the Application the „fs“ Package is used
- Works not really well ☹

Sending multi-media files, client side

- By submitting the formular, a function gets called wich checks if there are any files to send
- The function can decide if we have an image, video or audio file
- The function creates a URL of the file that will be sent to the Server
- By using the socket.emit function, there is an Event for each file type
- By Using the socket.on function, the Clients will get the File URL from the Server

Sending multi-media files, server side

- By using the socket.on function, the server gets the Event
- By using the io.emit function, the Server will sent the URL to the Clients in the Chatroom

Lessons learned

- Programming a Node.js Application with Frontend
  - *Sending html Files to Client*
  - *Routing those files in Node.js*
- Working with socket.io
- How to send Files from Client to Server and back
- Starting earlier to make UML Diagrams
- Project Management

Component Diagram