#1 | Chat messages don't seem to support new lines   [BUG]

Functional Requirements
- A message that has been submitted with new lines don't show up in rendering in the chat

Technical Requirements
- New lines are given by the backend with \n characters, so the frontend need to convert these to newlines in the UI

#2 | Replace dummy chat messages with data from the API   [FEATURE]

Functional Requirements
- Currently we have dummy data in place in the Chat UI, this should be replaced with live data from the server

Technical Requirements
- Connect the data from the API available on `[GET] /messages`

#3 | Create Messages API to show chat messages in the Chat UI [FEATURE]

Technical Requirements
- Create an API on `[GET] / messages` which returns a JSON response with the following fields:

```
{
  "message": String,
  "person": {
    "name": String,
    "avatar": String in format URL,
  },
  "datetime": String in format ISO 8601 with timezone
}
```

_____

#4 | Make sure /healthz response returns 204   [BUG]

Technical Requirements
- The health endpoint currently returns a Http Status 200 with nonsense response body. Make sure the body is clear and endpoint returns status 204.

_____

#5 | Add new message to the chat   [FEATURE]

Functional Requirements
- When clicking the Plus icon in the top right corner, a modal should open where you can type in a new message.
- You should be able to write a multiline message
- You should be able to submit the message by clicking a submit button.
- Pressing enter should allow you to add a new line in the multiline editor and not submit the message

Technical Requirements
- Create an API on `[POST] /messages` where new messages can be posted to.

_____

#6 | Hide Call button from Contacts when no phone number   [FEATURE]
is given

Functional Requirements
- Some contacts don't have a phone number filled in, so hide the call button when it's missing.

#7 | Replace fixed contacts in UI with data from Contacts API   [FEATURE]

Technical Requirements
- Connect the Contacts list in the frontend with the Contacts API endpoint.
- Make sure all phone numbers are formatted correctly so they won't break the call button
- Use the Contacts model which already contains the contact list
- Create an API on `[GET] /contacts` which returns a JSON response with the following fields:

```
{
  "name": String,
  "avatar": String,
  "phone": String,
  "email": String,
}
```

#8 | Add option to only show your favourite contacts in the sidebar [FEATURE]

Functional Requirements
- You should be able to switch between all contacts and your favourite contacts
- It should be clear in the UI which contact is favourite

Technical Requirements
- Add the option to contacts in the Contacts API to mark contacts as favourite

---

#9 | Sort contacts by last name   [FEATURE]

Functional Requirements
- All contacts should be sorted by last name