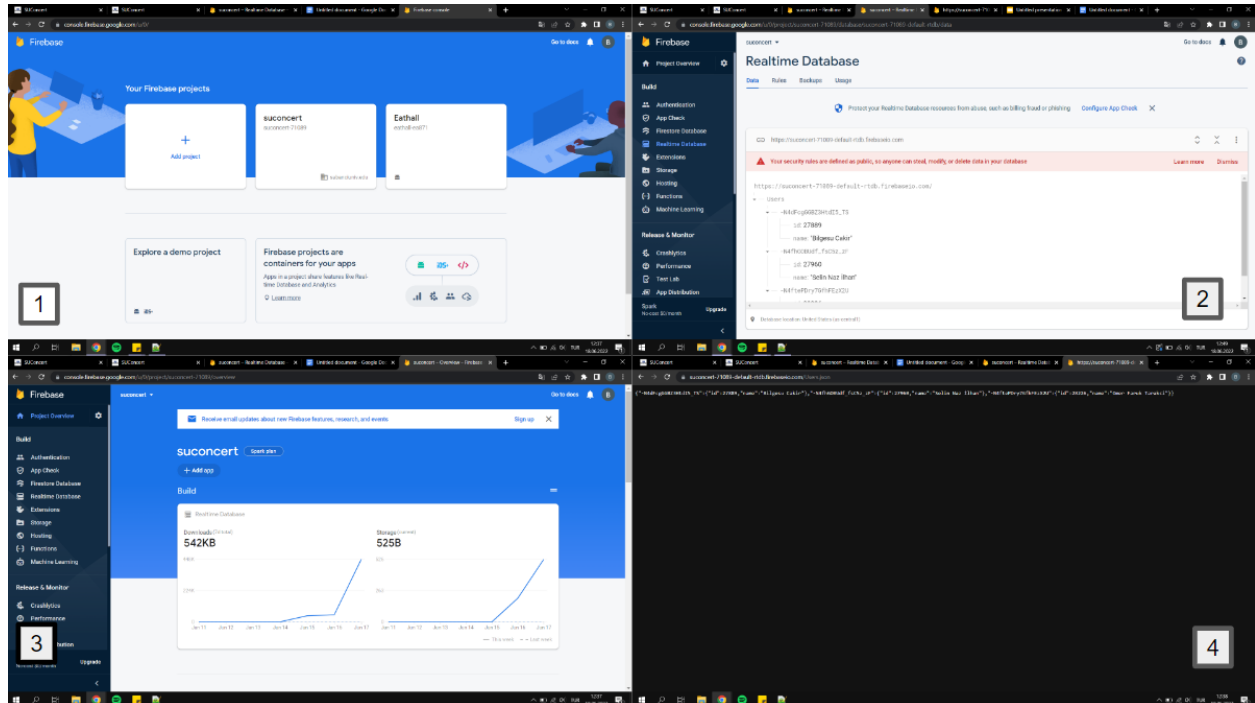# CS306 Project Step 4

Bilgesu Çakır 27889

Selin Naz İlhan 27960

The first step of the final part of our project was to create a firebase that stored all of the chat that will be sent in our support page. We created a "suconcert" project in firebase (1). We started using the firebase(3) for realtime database during this project step(2). First we set out to understand how json works in a firebase. We created our URL "https://suconcert-71089-default-rtdb.firebaseio.com/" and sent a demo "Users" page using rest ninja(4). Then we started out with the chat page, seeing the users page sent the data accordingly. To make sure we don't miss any steps during this report we deleted our chats, which we will now re-implement while screenshotting the whole process.

We started by adding a Live Chat button to our database. This link is clickable and redirects us to "live_chat.php" which we will include in the zip file during our submission. The code block for the redirection is attached below. This is added to the "main_page.php" as can be seen from above.

```
<tr align="center">
    <td align="center">
        <table align="center" width="100%">
            <tr align="center">
                <td align="center">

                    For help you can use <a href="http://localhost/suconcert/live_chat.php">Live Chat</a>

                </td>
            </tr>
        </table>
    </td>
</tr>
```
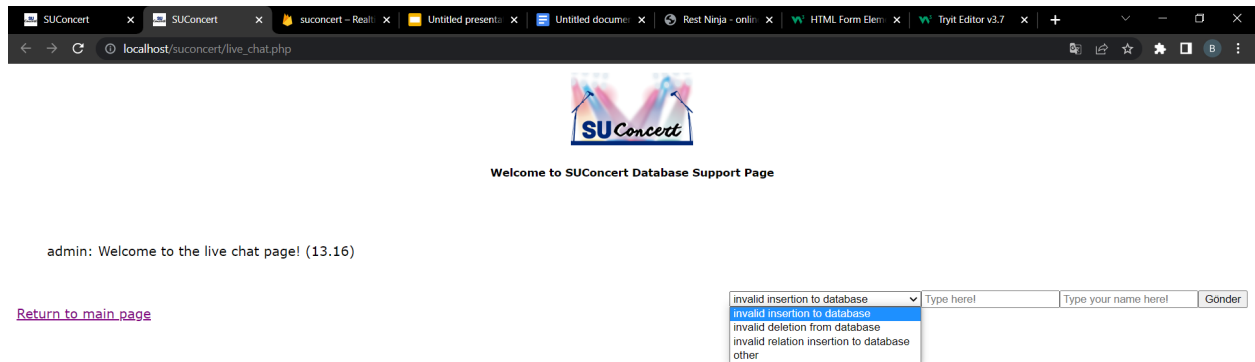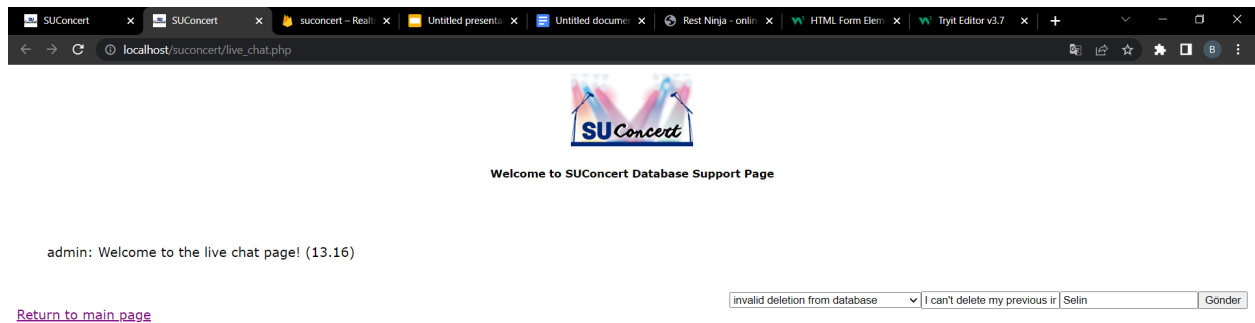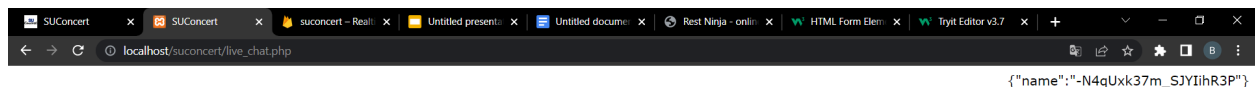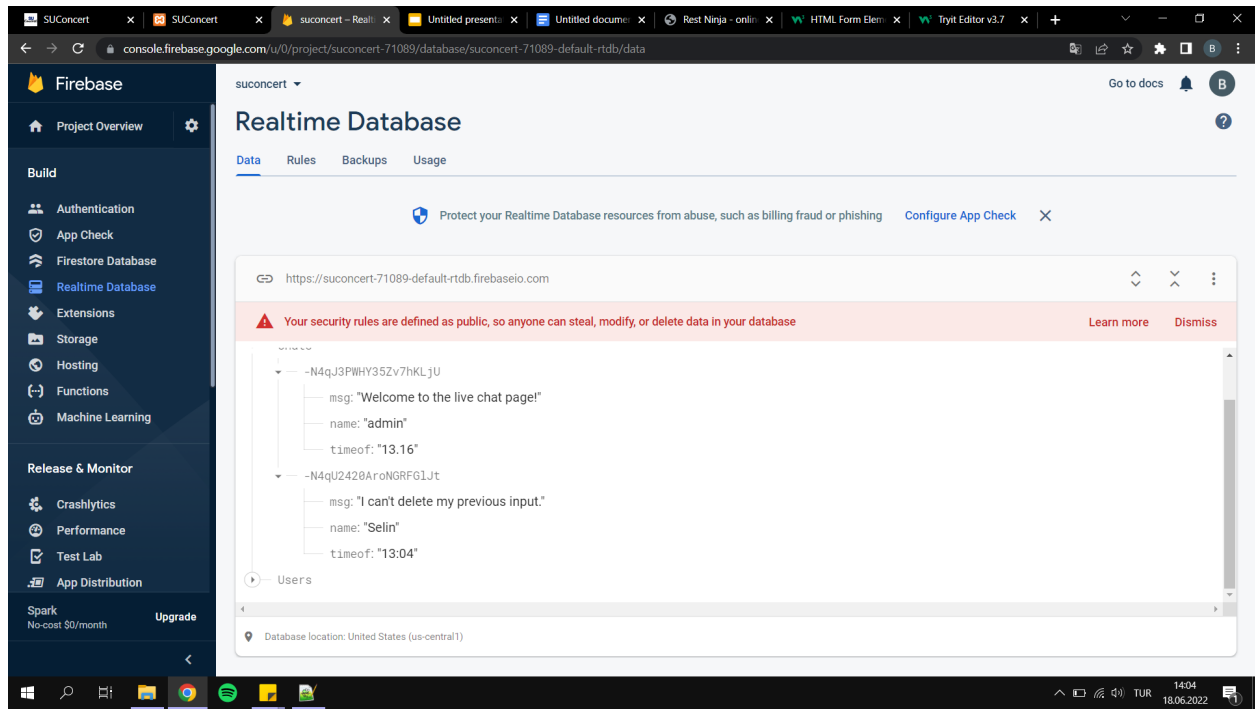
Then we go into the live chat.

We created admin manually using rest ninja. We also eliminated rest ninja's use further on with our admin chat page, which we describe in page 6. Although admin is created with rest ninja, we send guest's messages to the database using our own database, and every guest needs to enter theri own name. There's a return to main page button, ant to its right, we have three boxes. The first box is the dropdown menu in order to understand why the user is currently ussing the live chat. The options for this dropdown can be seen below.

Let's say User Selin has a problem with the deletion function.

We see that realtime database recognizes Selin and her message, and we see in the live chat page that the messages is received.

We added a line and the subject below in order to not forget what the chosen subject is about. We will now show a conversation in order to make sure our live_chat works appropriately. Then, we will add the screenshots from our database as well as the .json file.

Welcome to SUConcert Database Support Page

admin: Welcome to the live chat page! (13.16)

selin: I can't delete my previous input. (13:08)

admin: Have you made sure that you selected the right input? (14.10)

selin: yes, but it didn't redirect me to any page. (13:11)

admin: Did you see result=1? (14.11)

selin: no, i didnt. (13:12)

admin: Try deleting the value you have again, there might be a connection error. Wheny ou see result=1, you'll understand that your deletion is succesful. (14.12)

admin: You can also check the database table below to see if your deletion passed through. (14.13)

selin: Thank you so much for your help! (13:14)

Return to main page

[invalid insertion to database ▼] [Type here!] [Type your name here!] [Gönder]

Firebase

suconcert ▾

Go to docs

Project Overview

Realtime Database

Build

Data   Rules   Backups   Usage

Authentication
App Check
Firestore Database
Realtime Database
Extensions
Storage
Hosting
Functions
Machine Learning

Protect your Realtime Database resources from abuse, such as billing fraud or phishing   Configure App Check   ×

https://suconcert-71089-default-rtdb.firebaseio.com

-N4qW9tbULFFlX23Tt7B
    msg: "Try deleting the value you have again, there might be a connection error. Wheny ou see result=1, you'll understand that your deletion is succesful."
    name: "admin"
    timeof: "14.12"

-N4qWGeZvTHcSmpM305U
    msg: "You can also check the database table below to see if your deletion passed through."
    name: "admin"
    timeof: "14.13"

-N4qWL87ztoqM453m6Th
    msg: "Thank you so much for your help!"
    name: "selin"
    timeof: "13:14"

Release & Monitor

Crashlytics
Performance
Test Lab
App Distribution

Spark
No-cost $0/month   Upgrade

Database location: United States (us-central1)

There is just one slight problem which we can't solve, which is the time stamp for the guest since it takes it's value not from our local computer but elsewhere, The timestamp is automatic, although it's stored on the firebase, we didn't want to manually change the data we received through our chat page, so we left it the way it is.

One final addition to our database live_chat was for us admins to follow through on all the issues of the guests and users so we also created a admin chat page, where the admin responds to the questions asked by guests. This way, we eliminated the used of rest ninja totally. The screenshots for admin chat is also below.