

**TEST PROJECT**

**EVENT MANGEMENT SYSTEM**

**(Until: 19.03.2021)**

(Report)

Name	Aycha yahia
ID	37958317126

## 1. PROJECT NAME:

### FULL-STACK EVENT MANAGEMENT LIST SYSTEM

## 2. ABSTRACT:

Event management list system can perform the following:

- CRUD operations over events included in events list.
- Provide the ability of login by password and email.

## 3. FRONT-END BY REACT:

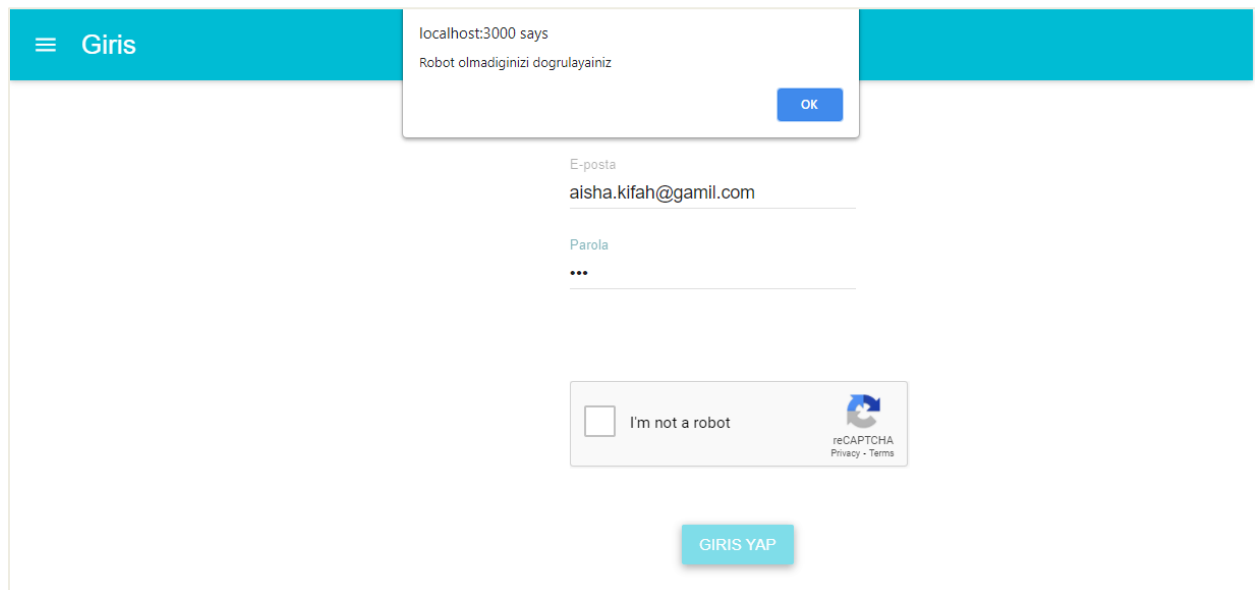
### 3.1 Log in interface:

Log interface sends GET request to back-end with encoded email and password param to control login information's. Based on backend responses have the following probabilities:

- Alert the user if the password and email do not match.
- Alert the user if the email is not registered in database.

Also login communicate with reCAPTCHA server to verify the user's response. (This communication accomplishes by the secret key)

- Alert if reCAPTCHA not verified.



The screenshot displays a web application's login interface. At the top left, there is a teal header with a hamburger menu icon and the text "Giris". A white modal box is open in the center, displaying a message from "localhost:3000" stating "Robot olmadiginizi dogrulayiniz" (Verify you are not a robot) with an "OK" button. Below the modal, the login form contains two input fields: "E-posta" (Email) with the value "aisha.kifah@gamil.com" and "Parola" (Password) with three dots indicating a masked password. Below these fields is a reCAPTCHA widget with the text "I'm not a robot" and a checkbox. At the bottom center of the form is a teal button labeled "GIRIS YAP".

**Figure 3.1.1** Login interface – reCAPTCHA not verified warning

≡

Giris

localhost:3000 says  
Sifre veya email yanlistir


OK

E-posta  
aisha.klfah@gamil.com

Parola  
....

✓

I'm not a robot

  
reCAPTCHA  
Privacy - Terms

GIRIS YAP

**Figure 3.1.2** Login interface – password and email not match warning

≡

Giris

localhost:3000 says  
Bu e-posta kayitlarimizda yok


OK

E-posta  
aisha.klffah@gamil.com

Parola  
....

✓

I'm not a robot

  
reCAPTCHA  
Privacy - Terms

GIRIS YAP

**Figure 3.1.3** Login interface – email not registered warning

### 3.2 Event Management List interface:

Sends GET request to back-end in order to get all lists in database and list received events over a table.

-Forward for add\_new\_event interface.

-Forward for update/details \_event interface.

-Sends Delete request to back-end with selected event id as path variable. (alerts before deleting.)

Etkinlik Yontem Sistemi

Etkinlikler List

Etkinlik Ekle

Takvim Goster

Etkinlik ID	Etkinlik baslik	Etkinlik icerik	Etkinlik durum	Etkinlik tarih	Etkinlik detay	Action
31	weeking1	teknolojy zervesi	pasif	2021-12-11T21:00:00.000+00:00	online ortamda düzenlenecek..	<div>Detay</div> <div>Sil</div>
32	WEEKENF3	ROBOTIK ZERVESI	aktif	2021-12-03T21:00:00.000+00:00	C 2012 SINFINDA düzenlenecek.	<div>Detay</div> <div>Sil</div>

All Rights Aycha yahia

Figure 3.2.1 event list interface

Etkinlik Yontem Sistemi

localhost:3000 says  
31'li olan etkinligi silmek istediginizden emin misiniz?

OK

Cancel

Etkinlik Ekle

Takvim Goster

Etkinlik ID	Etkinlik baslik	Etkinlik icerik	Etkinlik durum	Etkinlik tarih	Etkinlik detay	Action
31	weeking1	teknolojy zervesi	pasif	2021-12-11T21:00:00.000+00:00	online ortamda düzenlenecek..	<div>Detay</div> <div>Sil</div>
32	WEEKENF3	ROBOTIK ZERVESI	aktif	2021-12-03T21:00:00.000+00:00	C 2012 SINFINDA düzenlenecek.	<div>Detay</div> <div>Sil</div>

All Rights Aycha yahia

Figure 3.2.2 event list interface- event deletion warning

### 3.3 Add event interface:

View text fields to fill all information's needed for an event. Sends POST request to back-end in order to insert a new event to the database. Whenever save button is clicked the interface sends POST request to back-end in order to apply changes over the selected event.

Alerts for:

-Null text field.

-Wrong date format.

The figure displays two side-by-side screenshots of a web application interface for adding a new event. Both screenshots show a form with the following fields: 'WEE' (with a dropdown menu), 'Icerik' (with a text input field), 'Durum' (with a dropdown menu), 'Tarih' (with a date input field), and 'Detay' (with a rich text editor). The 'Detay' field contains the text 'SALON 2.2 C'DE OLACAK'. At the bottom of the form, there are two buttons: 'Ekle' (green) and 'Iptal et' (red).

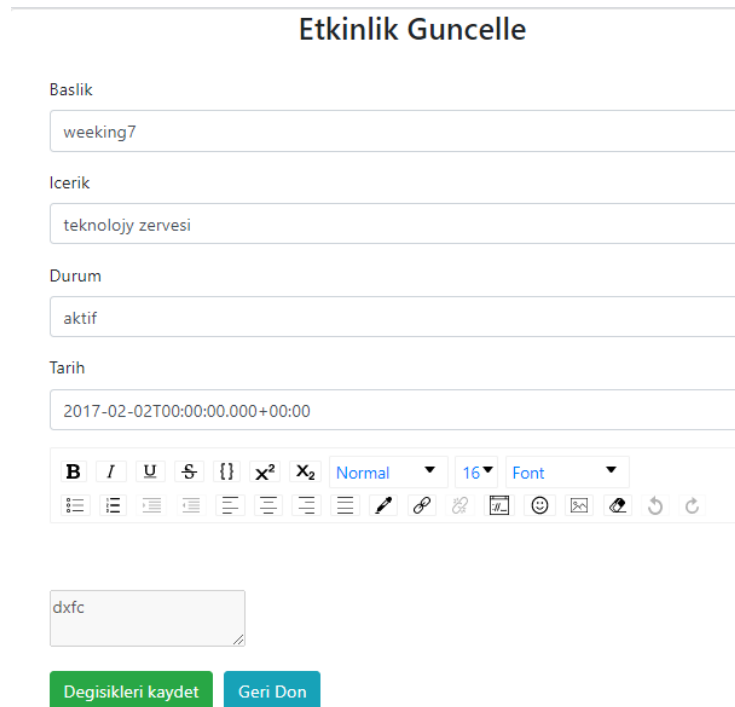
**Left Screenshot (Null text field warning):** A warning dialog box is displayed over the 'Icerik' field. The dialog box contains the text 'localhost:3000 says' and 'icerik bos olamaz!' (content cannot be empty). The 'OK' button is visible.

**Right Screenshot (Wrong date format warning):** A warning dialog box is displayed over the 'Tarih' field. The dialog box contains the text 'localhost:3000 says' and 'Tarih Formati bu sekildedir : yyyy-MM-dd' (Date format is like this: yyyy-MM-dd). The 'OK' button is visible.

**Figure 3.3.1** add new event interface- Null text field, Wrong date format warning

### 3.4 Update event interface:

View a selected event in individual interface. Takes the id sent by event list interface via the url and send GET request to the back-end to get the specified event from the database. Whenever update button is clicked the interface sends POST request to back-end in order to apply changes over the selected event.



**Etkinlik Guncelle**

Baslik  
weeking7

Icerik  
teknoloji zervesi

Durum  
aktif

Tarih  
2017-02-02T00:00:00.000+00:00

**B** *I* U ~~S~~ `{ }`  $x^2$   $x_2$  Normal 16 Font

dxfc

Degisikleri kaydet Geri Don

**Figure 3.4.1** add new event interface- Null text field, Wrong date format warning

### 5. Problems and errors faced:

- 1- There was a problem with running spring boot application. This error occurred because of the compiler was not able to scan Controller package. The reason of this error was the order of packages in project architecture. From the document it is concluded that generally it is recommend to locate main application class in a root package above other classes. The @EnableAutoConfiguration annotation is often placed on main class, and it implicitly defines a base “search package” for certain items. Using a root package also allows the @ComponentScan annotation to be used without needing to specify a basePackage attribute.
- 2- 400 bad request error: This error occurred while trying to make post request from client to server. After goggling it, it turns out that this was happening because of the request build sent by client.

### 6. Not completed parts and technique problems:

- 1- Calendar code is not preformed.
- 2- Date is just accepted in this format :  
Yyyy-mm-dd, no ability to add the time. The problem is exactly occurs while control the format of input date. The ready function isDate() is used in front-end to control the format of the date before being sent to back-end. isDate() returns true for dates in yyyy-mm-dd else returns false. Without using this validity control the user will be able to use YYYY-MM-DD hh:mm:ss. But if the user makes any mistake backend will return exception and user will not be able to know the problem.

- 3- In update interface, I was not able to show “detay” value of the event on the editor. So I added text area. I am viewing the original “ detay” and edited “detay” values at the same time. But only editor value will be added to database. So the user first has to copy content of the text area to editor then edit if he do not to lose the whole content.