SPRINT – 3

Date	**12/10/2022**
Team ID	**PNT2022TMID32597**
Project Name	**Personal Expense Tracker**

EXPENSE :

```
import React from "react"
import Swal from "sweetalert2"

export default function expensePage(email, balance, func) {
    // render() {
    const alertUser = async () => {
        const { value: formValues } = await Swal.fire({
            title: "Add Expense",
            html: '<input id="amount" class="swal2-input" placeholder="Enter
amount" type="number">' +
                '<select name="category" class="swal2-input" id="category"
><option value="" disabled selected>Select category</option><option
value="grocerry">Grocerry</option><option value="utility bills">Utility
Bills</option><option value="transport">Transport</option><option
value="education">Education</option><option
value="medicine">Medicine</option><option
value="others">Others</option></select>',
            position: 'center',
            showConfirmButton: true,
            showCancelButton: true,
            showCloseButton: true,
            timerProgressBar: false,
            confirmButtonText: "Add",
            confirmButtonColor: "green",
            preConfirm: () => {
                return [
                    document.getElementById('amount').value,
                    document.getElementById('category').value
                ]
            }
        })

        if (formValues) {
            if (formValues[0] === '' || formValues[1] === '' ||
parseInt(formValues[0])<= 0) {
```

```
            Swal.fire({
                title: "Invalid Expense",
                text: "Please provide both values",
                position: 'center',
                showConfirmButton: false,
                icon: 'error',
                toast: true,
                timerProgressBar: true,
                timer: 2000
            })
        } else {
            const url = new URL("http:localhost:5000/addExpense")
            let expense = {
                amount: parseInt(formValues[0]),
                category: formValues[1],
                email: email
            }
            const res = await fetch(url, {
                method: 'POST',
                headers: {
                    'Content-Type': 'application/json'
                },
                body: JSON.stringify(expense)
            })
            const data = await res.json()
            if (data.status === 200) {
                Swal.fire({
                    text: "Successfully Added",
                    position: 'top-right',
                    showConfirmButton: false,
                    icon: 'success',
                    toast: true,
                    timerProgressBar: true,
                    timer: 2000
                })
                if (balance - parseInt(formValues[0]) < 0) {
                    let url = new URL("http://localhost:5000/limitExceed")
                    url.searchParams.set('email', email)
                    fetch(url).then((res) => {
                        console.log(res.json())
                    })
                }
            } else {
                Swal.fire({
                    text: "Error ocuured",
```

```
                        position: 'center',
                        showConfirmButton: false,
                        icon: 'error',
                        toast: true,
                        timerProgressBar: true,
                        timer: 2000
                    })
                }
                func()
            }
        }
    }

    return (<>
        {alertUser()}
    </>)

    // }
}
```

SEND GRID :

```
import os
from sendgrid.helpers.mail import Mail
from sendgrid import SendGridAPIClient

# from address we pass to our Mail object, edit with your name
FROM_EMAIL = 'balasaravanan150702@gmail.com'

# update to your dynamic template id from the UI
TEMPLATE_ID = 'd-4d87db5df1564451a65390ca17cbd056'

# list of emails and preheader names, update with yours
TO_EMAILS = [('19euit050@skcet.ac.in', 'Krish'),('19euit051@skcet.ac.in',
'Krish')]

API_KEY =""
def SendDynamic():
    """ Send a dynamic email to a list of email addresses

    :returns API response code
    :raises Exception e: raises an exception """
    # create Mail object and populate
```

```
    message = Mail(
        from_email=FROM_EMAIL,
        to_emails=TO_EMAILS)
    # pass custom values for our HTML placeholders
    message.dynamic_template_data = {
        'subject': 'Expense Limit Exceed Alert',
        # 'place': 'New York City',
        # 'event': 'Twilio Signal'
    }
    message.template_id = TEMPLATE_ID
    # create our sendgrid client object, pass it our key, then send and return
our response objects
    try:
        sg = SendGridAPIClient(API_KEY)
        global response
        response = sg.send(message)

        code, body, headers = response.status_code, response.body,
response.headers
        print(f"Response code: {code}")
        print(f"Response headers: {headers}")
        print(f"Response body: {body}")
        print("Dynamic Messages Sent!")
    except Exception as e:
        print("Error: {0}".format(e))
    return str(response.status_code)

if __name__ == "__main__":
    SendDynamic()
```