

IMPLEMENTING WEB APPLICATION

Date	10 November 2022
Team ID	PNT2022TMID35580
Project Name	Project – Personal Expense Tracker Application

Implementing Various modules

App.py

```
from flask import Flask,render_template,request,redirect,url_for,session
from datetime import date
import ibm_db
import bcrypt
import re
import json
import os
from sendgrid import SendGridAPIClient
from sendgrid.helpers.mail import Mail

app=Flask(__name__)
app.secret_key='a'

conn=ibm_db.connect("DATABASE=bludb;HOSTNAME=b70af05b-76e4-4bca-a1f5-
23dbb4c6a74e.clogj3sd0tgtu0lqde00.databases.appdomain.cloud;PORT=32716;Securit
y=SSL;SSLServerCertificate=DigiCertGlobalRootCA.crt;UID=dgp98488;PWD=T4ZhPsFfV
gztg07H;", "", "")

@app.route('/')
def home():
    return render_template('signup.html')

@app.route('/dashboard')
def dash():
    return render_template('dashboard.html')

@app.route('/graph')
def graph():
    return render_template('graph.html')

@app.route('/display')
def display():
    print (session["email"],session["id"])

@app.route("/api/login",methods=['POST'])
def login():
    msg={};
    if request.method=="POST":
```

```

        email=request.form['email']
        password=request.form['pw']
        sql="SELECT * from test where email=?;"
        stmt=ibm_db.prepare(conn,sql)
        ibm_db.bind_param(stmt,1,email)
        ibm_db.execute(stmt)
        account=ibm_db.fetch_assoc(stmt)
        if account and bcrypt.checkpw(password.encode('utf-8'),
account["PASSWORD"].encode('utf-8')):
            session['Loggedin']=True
            session['id']=account['USERID']
            session['email']=account['EMAIL']
            session['name']=account['USERNAME']
            msg={"Success": True}
        else:
            msg={"Success": False, "msg": "Account does not exist or wrong
email/password! Kindly Check!"}
        return json.dumps(msg)

@app.route("/api/register",methods=["POST"])
def register():
    msg={}
    if request.method=="POST":
        username=request.form['name']
        email=request.form['email']
        password=request.form['pass']
        hashed = bcrypt.hashpw(password.encode('utf-8'), bcrypt.gensalt())
        sql="SELECT * FROM test where username=?"
        stmt=ibm_db.prepare(conn,sql)
        ibm_db.bind_param(stmt,1,username)
        ibm_db.execute(stmt)
        account=ibm_db.fetch_assoc(stmt)
        if account:
            msg={"Success": False, "msg": "Account already exists!!"}
        elif not re.match(r'^@+@[^@]+\.[^@]+',email):
            msg={"Success": False, "msg": "Invalid Email Address!"}
        elif not re.match(r'([a-zA-Z]|[0-9])+',username):
            msg={"Success": False, "msg": "Name must contain only characters
and numbers!"}
        else:
            insert_sql="INSERT into test(email, username, password) values
(?,?,?);"
            prep_stmt=ibm_db.prepare(conn,insert_sql)
            ibm_db.bind_param(prepare_stmt,1,email)
            ibm_db.bind_param(prepare_stmt,2,username)
            ibm_db.bind_param(prepare_stmt,3,hashed)
            ibm_db.execute(prepare_stmt)

```

```

        msg={"Success": True, "msg": "You have registered for the App
successfully!"}
        return json.dumps(msg)

@app.route("/api/expenses", methods=["POST"])
def add_expense():
    if "id" in session.keys():
        user = session["id"]
        date_now = request.form["date"]
        amount = request.form["amount"]
        category = request.form["category"]
        insert_sql="INSERT into expenses(user, date, amount, category) values
(?,?,?,?);"
        prep_stmt=ibm_db.prepare(conn,insert_sql)
        ibm_db.bind_param(prepare_stmt,1,user)
        ibm_db.bind_param(prepare_stmt,2,date_now)
        ibm_db.bind_param(prepare_stmt,3,amount)
        ibm_db.bind_param(prepare_stmt,4,category)
        msg = {"msg": f"{ibm_db.execute(prepare_stmt)}",}
        print("Calling Send...")
        send(user, session["email"])
        return json.dumps(msg);
    return json.dumps({"msg": "Invalid session: Log in"});

@app.route("/api/expenses", methods=["GET"])
def view_expense():
    if "id" in session.keys():
        user = session["id"]
        sql_stmt="SELECT * from expenses where expenses.user=?"
        stmt=ibm_db.prepare(conn,sql_stmt)
        ibm_db.bind_param(stmt,1,user)
        ibm_db.execute(stmt)
        row = ibm_db.fetch_assoc(stmt)
        ret = []
        while row != False:
            ret.append({"user":row["USER"], "amount": -row["AMOUNT"],
"category": row["CATEGORY"], "date": f"{row['DATE']}"})
            row = ibm_db.fetch_assoc(stmt)
        return json.dumps(ret)
    return json.dumps({"msg": "Invalid session: Log in"});

@app.route("/api/expenses_by_category", methods=["GET"])
def view_expense_by_catargory():
    if "id" in session.keys():
        user = session["id"]
        sql_stmt="SELECT category, SUM(amount) as amount from expenses where
expenses.user=? and amount>0 group by category"
        stmt=ibm_db.prepare(conn,sql_stmt)

```

```

        ibm_db.bind_param(stmt,1,user)
        ibm_db.execute(stmt)
        row = ibm_db.fetch_assoc(stmt)
        ret = {"food":0,"entertainment":0,"shopping":0, "emi": 0, "rent":0,
"others": 0}
        while row != False:
            ret[row["CATEGORY"]]= row["AMOUNT"]
            row = ibm_db.fetch_assoc(stmt)
        return json.dumps(ret)
    return json.dumps({"msg": "Invalid session: Log in"});

@app.route("/api/balance", methods=["GET"])
def view_balance():
    if "id" in session.keys():
        user = session["id"]
        sql_stmt="SELECT SUM(amount) as balance from expenses where
expenses.user=?"
        stmt=ibm_db.prepare(conn,sql_stmt)
        ibm_db.bind_param(stmt,1,user)
        ibm_db.execute(stmt)
        row = ibm_db.fetch_assoc(stmt)
        ret = {};
        if row != False and row["BALANCE"]!=None:
            ret["balance"]= -row["BALANCE"];
        else:
            ret["balance"]= 0;
        return json.dumps(ret)
    return json.dumps({"msg": "Invalid session: Log in"});

@app.route("/api/session", methods=["GET"])
def get_session():
    if "id" in session.keys():
        ret = {}
        ret['id']=session['id']
        ret['email']=session['email']
        ret['name']=session['name']
        return json.dumps(ret)
    return json.dumps({"msg": "Invalid session: Log in"});

@app.route("/api/session", methods=["DELETE"])
def clear_session():
    session.clear();
    return json.dumps({"Success":True})

def send(user, email):
    print(email)
    sql_stmt="SELECT SUM(amount) as balance from expenses where
expenses.user=?"

```

```

stmt=ibm_db.prepare(conn,sql_stmt)
ibm_db.bind_param(stmt,1,user)
ibm_db.execute(stmt)
row = ibm_db.fetch_assoc(stmt)
ret = {};
if row != False and row["BALANCE"]!=None and row["BALANCE"]>0:
    message=Mail(from_email='ramyanmsr@gmail.com',
                  to_emails=email,
                  subject='ALERT!! Budget Exceeded!',
                  plain_text_content='Budget has been crossed for this
month',
                  html_content='<strong>Hi user,<br>You have crossed your
budget for this month. Its a remainder for you to start saving!</strong>')
    try:
        sg = SendGridAPIClient(os.environ.get('SENDGRID_API_KEY'))
        print(os.environ.get('SENDGRID_API_KEY'))
        response=sg.send(message)
        print(response.status_code)
        print(response.body)
        print(response.headers)

    except Exception as e:
        print(e)

if __name__ == "__main__":
    app.run(host="0.0.0.0", debug=True)

```

- **Login and Registration**

Source Code:

```

<!DOCTYPE html>
<html lang="en" dir="ltr">
  <head>
    <meta charset="UTF-8">
    <title>Personal Expense Tracker Login</title>
    <link rel="stylesheet" href="{{ url_for('static',
filename='css/style.login.css') }}" />
    <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/font-
awesome/5.15.3/css/all.min.css" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0"
/>
  </head>
  <body>
    <div class="container">
      <input type="checkbox" id="flip">
      <div class="cover">

```

```

        <div class="front">
            <div class="text">
            </div>
        </div>
        <div class="back">
            <div class="text">
            </div>
        </div>
    </div>
    <div class="forms">
        <div class="form-content">
            <div class="login-form">
                <div class="title">Login</div>
                <form onsubmit="submitLoginForm(event)">
                    <div class="input-boxes">
                        <div class="input-box">
                            <i class="fas fa-envelope"></i>
                            <input type="text" id="email" name="email"
placeholder="Enter your email" required>
                        </div>
                        <div class="input-box">
                            <i class="fas fa-lock"></i>
                            <input type="password" id="pw" name="pw"
placeholder="Enter your password" required/>
                            <i onclick="toggleShowPass(event)" style =
"padding-left:330px;"class="fa fa-eye"></i>
                        </div>

                        <div class="button input-box">
                            <input type="submit" value="Submit">
                        </div>
                        <div class="text sign-up-text">Don't have an
account? <label for="flip">Signup now</label></div>
                    </div>
                </form>
            </div>
            <div class="signup-form">
                <div class="title">Signup</div>
                <form onsubmit="submitRegisterForm(event)">
                    <div class="input-boxes">
                        <div class="input-boxes">
                            <div class="input-box">
                                <i class="fas fa-user"></i>
                                <input type="text" id="name" name="name"
placeholder="Enter your name" required>
                            </div>
                            <div class="input-box">
                                <i class="fas fa-envelope"></i>

```

```

        <input type="text" id="email" name="email"
placeholder="Enter your email" required>
    </div>
    <div class="input-box">
        <i class="fas fa-lock"></i>
        <input type="password" id="pass"
name="pass" placeholder="Enter your password" required>
        <i onclick="toggleShowPass(event)" style =
"padding-left:330px;"class="fa fa-eye"></i>
    </div>
    <div class="button input-box">
        <input type="submit" value="Submit">
    </div>
    <div class="text sign-up-text">Already have an
account? <label for="flip">Login now</label></div>
    </div>
</form>
</div>
</div>
</div>
<script type="text/javascript" src="{ { url_for('static',
filename='js/script.login.js') }}"></script>
</body>
</html>

```

Java Script file:

```

let toggleShowPass = (event) => {
    let x = event.currentTarget.parentElement.querySelector("input");
    if (x.type === "password") {
        x.type = "text";
    } else {
        x.type = "password";
    }
}

const submitLoginForm = (e) => {
    e.preventDefault()
    let form = event.currentTarget;
    fetch("http://localhost:5000/api/login", {
        method: "POST",
        body: new FormData(form)
    })
    .then(body=>body.json())
    .then((x)=>{
        if (!x.Success){
            alert(x.msg);

```

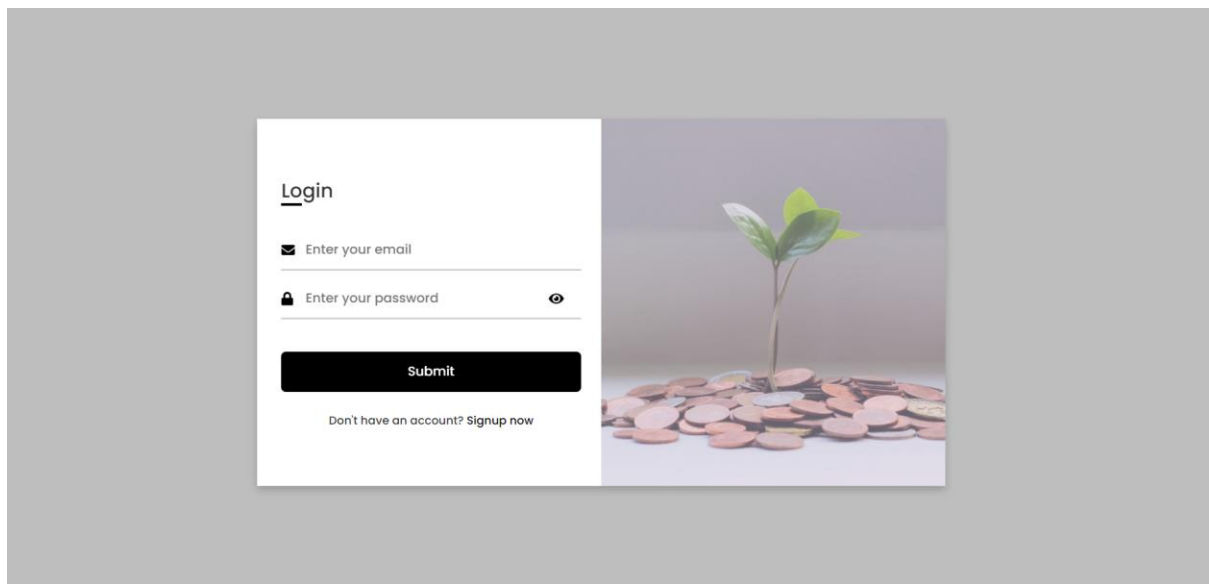
```

    } else {
      window.location = "http://localhost:5000/dashboard"
    }
  })
}

const submitRegisterForm = (e) => {
  e.preventDefault()
  let form = event.currentTarget;
  fetch("http://localhost:5000/api/register", {
    method: "POST",
    body: new FormData(form)
  })
  .then(body=>body.json())
  .then((x)=>{
    if (!x.Success){
      alert(x.msg);
    } else {
      window.location = "http://localhost:5000/"
    }
  })
}

```

Output:



Login

✉ ramyanmsr@gmail.com

🔒



Submit

Don't have an account? [Signup now](#)



Signup

👤 Enter your name

✉ Enter your email

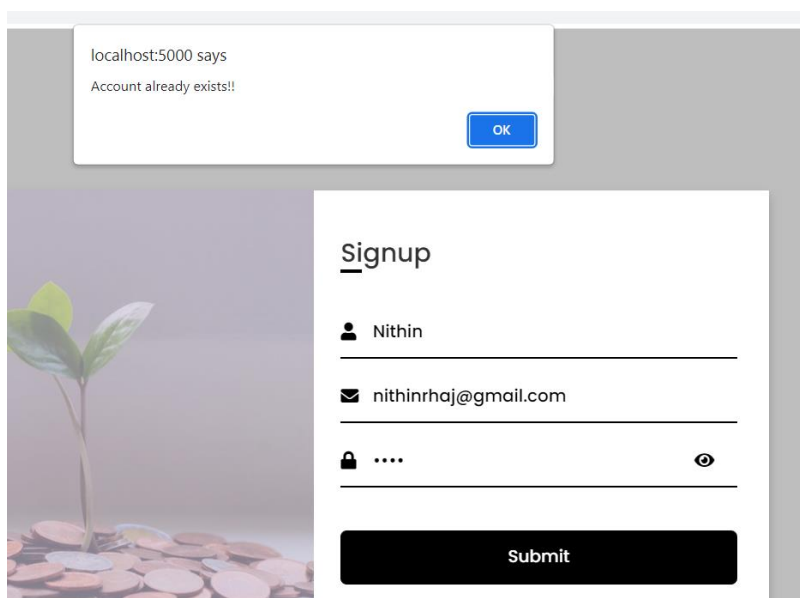
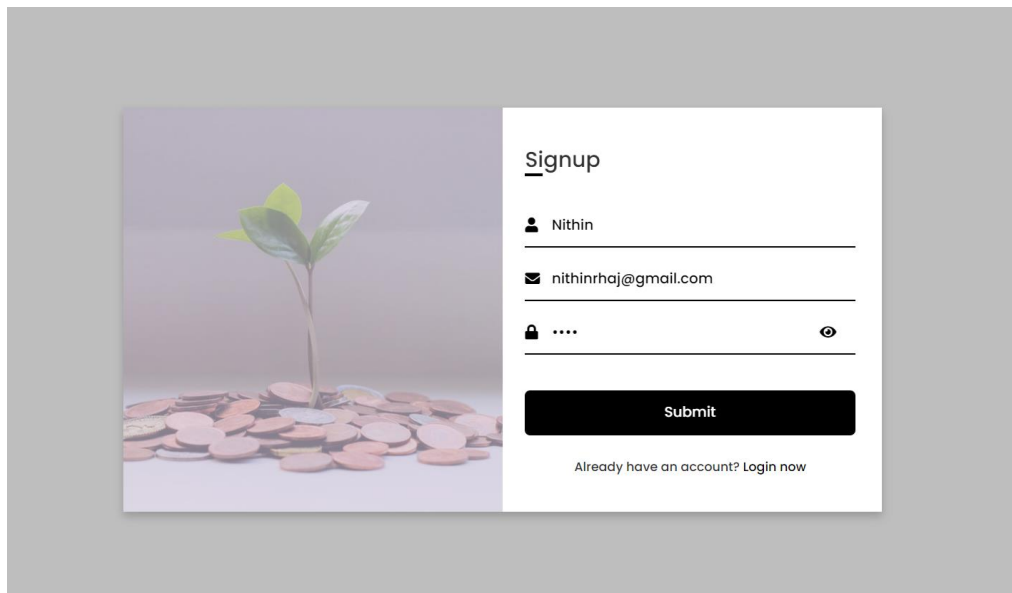
🔒 Enter your password



Submit

Already have an account? [Login now](#)





- **Dashboard – Add Money to wallet & Add Expenditures**

Source Code:

```
<!DOCTYPE html>
<html lang="en" >
  <head>
    <meta charset="UTF-8">
    <title>The Wallet</title>
    <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/meyer-reset/2.0/reset.min.css">
    <link rel="stylesheet" href="{ url_for('static',
filename='css/style.dashboard.css') }}">
  </head>
  <body onload="main()">
    <!-- partial:index.partial.html -->
    <div class="app">
```

```

        <header class="app-header">
            <div class="app-header-logo">
                <div class="logo">
                    <span class="logo-icon">
                        
                    </span>
                    <h1 class="logo-title">
                        <span>The Wallet</span>
                        <span>Expense Tracker</span>
                    </h1>
                </div>
            </div>
            <div class="app-header-navigation">
                <div class="tabs">
                    <marquee direction="right"><h1>If you are saving, you
are succeeding...</h1></marquee>
                </div>
            </div>
            <div class="app-header-actions">
                <button class="user-profile">
                    <span id="user_name">Loading...</span>
                    <span>
                        
                    </span>
                </button>
                <div class="app-header-actions-buttons">
                    <button class="icon-button large" onclick="logout()">
                        <i class="ph-sign-out"></i>
                    </button>
                </div>
            </div>
            <div class="app-header-mobile">
                <button class="icon-button large">
                    <i class="ph-list"></i>
                </button>
            </div>
        </header>
        <div class="app-body">
            <div class="app-body-navigation">
                <nav class="navigation">
                    <a href="/graph">
                        <i class="ph-browsers"></i>
                        <span>Dashboard</span>
                    </a>
                    <a href="/dashboard">
                        <i class="ph-check-square"></i>

```

```

        <span>Expenses</span>
    </a>
</nav>
<footer class="footer">
    <h1>The Wallet<small>@</small></h1>
</footer>
</div>
<div class="app-body-main-content">
    <section class="service-section">
        <h2>Expenses</h2>
        <!-- <div class="service-section-header">
            <div class="search-field">
                <i class="ph-magnifying-glass"></i>
                <input type="text" placeholder="Account
number">

            </div> -->
            <!-- For Joint account drop down based on account
number -->

            <!-- <div class="dropdown-field">
                <i class="ph-caret-down"></i>
            </div>
            <button class="flat-button">
                Search
            </button>
        </div>
        <div class="mobile-only">
            <button class="flat-button">
                Toggle search
            </button>
        </div> -->
        <div class="tiles">
            <article class="tile">
                <div class="tile-header">
                    <span>
                        
                    </span>
                    <h3>
                        <span><b> Food </b></span>
                    </h3>
                </div>
                <a href="#divOne">
                    <span>Add expenses</span>
                    <span class="icon-button">
                        <i class="ph-caret-right-bold"></i>
                    </span>
                </a>
            </article>

```

```

<article class="tile">
  <div class="tile-header">
    <span>
      
    </span>
    <h3>
      <span>Entertainment</span>
    </h3>
  </div>
  <a href="#divOne">
    <span>Add expenses</span>
    <span class="icon-button">
      <i class="ph-caret-right-bold"></i>
    </span>
  </a>
</article>
<article class="tile">
  <div class="tile-header">
    <span>
      
    </span>
    <h3>
      <span>Shopping</span>
    </h3>
  </div>
  <a href="#divOne">
    <span>Add expenses</span>
    <span class="icon-button">
      <i class="ph-caret-right-bold"></i>
    </span>
  </a>
</article>
<article class="tile">
  <div class="tile-header">
    <span>
      
    </span>
    <h3>
      <span>EMI</span>
    </h3>
  </div>
  <a href="#divOne">
    <span>Add expenses</span>
    <span class="icon-button">
      <i class="ph-caret-right-bold"></i>

```

```

        </span>
    </a>
</article>
<article class="tile">
    <div class="tile-header">
        <span>
            
        </span>
        <h3>
            <span>Rent</span>
        </h3>
    </div>
    <a href="#divOne">
        <span>Add expenses</span>
        <span class="icon-button">
            <i class="ph-caret-right-bold"></i>
        </span>
    </a>
</article>
<article class="tile">
    <div class="tile-header">
        <span>
            
        </span>
        <h3>
            <span>Others</span>
        </h3>
    </div>
    <a href="#divOne">
        <span>Add expenses</span>
        <span class="icon-button">
            <i class="ph-caret-right-bold"></i>
        </span>
    </a>
</article>
</div>
</section>
<section class="transfer-section">
    <div class="transfer-section-header">
        <h2>Recent Expenses</h2>
        <div class="filter-options">
            <p>Filter </p>
            <button class="icon-button"
onclick="reverseTransactions()">
                <i class="ph-funnel"></i>
            </button>
        </div>
    </div>

```

```

        </div>
    </div>
    <div class="transfers">
    </div>
</section>
</div>
<div class="app-body-sidebar">
    <section class="payment-section">
        <h2>Digital Wallet</h2>
        <div class="payment-section-header">
            <p>Saving, Investing and Tracking</p>
        </div>
        <div class="payments">
            <div class="payment">
                <div class="card green">
                    <span>&nbsp;</span>
                    <span>
                        
                    </span>
                    <span>
                        Balance
                    </span>
                </div>
                <div class="payment-details">
                    <span style="display: flex; justify-
content: space-between;">
                        <h3>Wallet Balance</h3>
                        <button onclick="openBalPopup()"
id="myBtn">Add to Wallet</button>
                    </span>
                </div>
                <span id="balance">Loading...</span>
            </div>
        </div>
    </div>
    <div class="faq">
        <h1>Frequently Asked Questions</h1>
        <div>
            <label>Do you lay a strong foundation for
long-term earnings and wealth?</label>
            <span style="color:#D1C1C1"
align="justify">Take every learning during college as your investment for the
future. Missing out on even one of those can seriously fritter away that
investment.</span>
        </div>
    </div>

```

```

                                <label>What financial mistakes should one
refrain from?</label>
                                <span style="color:#D1C1C1"
align="justify">Ensure that your under graduation loan amount should not be
more than your first year salary. Another mistake to avoid is playing with
credit card.</span>
                                </div>
                                <div>
                                <label>How to manage debt and boost your
credit score?</label>
                                <span style="color:#D1C1C1"
align="justify">Do not be influenced by your peers, but stick to maintaining
your own budget. Take up any campus employment to balance your expenditures
and make a saving, along with offering and gaining valued
experience. </span>
                                </div>
                                </div>
                                </div>
                                </section>
                                </div>
                                </div>
                                <div id="myModal" class="modal">
                                <!-- Modal content -->
                                <div class="modal-content">
                                <span class="close"
onclick="delParent(event)">&times;</span>
                                <form onsubmit="updateBalance(event)" style="color:black">
                                <h1 style="font-weight: bolder; font-size: x-
large;"><center>Modify Wallet Balance</center></h1>
                                <br/>
                                <div style="margin-left:185px; margin-top: 2px;">
                                <label for="amount" style="font-size:
large;"><b>Amount : </b></label>
                                <input type="number" style="margin-left:1px"
placeholder="Enter Amount" name="amount" required />
                                <button type="submit" class="btn">Add</button>
                                </div>
                                </form>
                                </div>
                                </div>
                                <div class="overlay" id="divOne">
                                <div class="wrapper">
                                <h2>Please Fill up The Expenses Form</h2><a class="close"
href="#">&times;</a>
                                <div class="content">
                                <div class="container">
                                <form onsubmit="addExpense(event)">
                                <label>Expense Category</label>

```



```

        <select name="category">
            <option value="food">Food</option>
            <option
value="entertainment">Entertainment</option>
            <option value="shopping">Shopping</option>
            <option value="emi">EMI</option>
            <option value="rent">Rent</option>
            <option value="others">Others</option>
        </select>
        <label>Date</label><br/>
        <input class="popup" placeholder="Date.."
type="date" required name="date">
        <label>Location</label>
        <input placeholder="Place of Purchase..."
type="text" class="popup">
        <label>Amount</label> <br/>
        <input class="popup" type="number"
placeholder="Enter Amount..." name="amount" required />
        <input class="popup" type="submit" value="Add
Expense">
    </form>
</div>
</div>
</div>
</div>
<!-- partial -->
<script src='https://unpkg.com/phosphor-icons'></script>
<script
src="https://unpkg.com/@babel/standalone/babel.min.js"></script>
<script type="text/babel" src="{ { url_for('static',
filename='/js/script.dashboard.js') } }"></script>
<script>
    window.watsonAssistantChatOptions = {
        integrationID: "8f7293cb-e55e-49e7-a937-915057242eca", //
The ID of this integration.
        region: "au-syd", // The region your integration is hosted
in.
        serviceInstanceID: "68106b4c-302a-41d4-bdf5-00458ba09a25",
// The ID of your service instance.
        onLoad: function(instance) { instance.render(); }
    };
    setTimeout(function(){
        const t=document.createElement('script');
        t.src="https://web-
chat.global.assistant.watson.appdomain.cloud/versions/" +
(window.watsonAssistantChatOptions.clientVersion || 'latest') +
"/WatsonAssistantChatEntry.js";
        document.head.appendChild(t);

```

```

    });
  </script>
</body>
</html>

```

Java Script File to work as API:

```

/** @jsx createElement */
/** @jsxFrag createFragment */

const createElement = (tag, props, ...children) => {
  if (typeof tag === "function") return tag(props, ...children);
  const element = document.createElement(tag);

  Object.entries(props || {}).forEach(([name, value]) => {
    if (name.startsWith("on") && name.toLowerCase() in window)
      element.addEventListener(name.toLowerCase().substr(2), value);
    else element.setAttribute(name, value.toString());
  });

  children.forEach(child => {
    appendChild(element, child);
  });

  return element;
};

const appendChild = (parent, child) => {
  if (Array.isArray(child))
    child.forEach(nestedChild => appendChild(parent, nestedChild));
  else
    parent.appendChild(child.nodeType ? child :
document.createTextNode(child));
};

const createFragment = (props, ...children) => {
  return children;
};

const displayBalance = () => {
  let val = document.querySelector("#balance")
  fetch("http://localhost:5000/api/balance")
    .then(body=>body.json())
    .then((x)=>{
      val.innerHTML=x.balance;
    })
}

```

```

}

const displayName = () => {
  let val = document.querySelector("#user_name")
  fetch("http://localhost:5000/api/session")
  .then(body=>body.json())
  .then((x)=>{
    if ("name" in x){
      val.innerHTML=x.name;
    } else {
      window.location = "http://localhost:5000"
    }
  })
}

const Transaction = (props) => (
  <div class="transfer">
    <div class="transfer-logo">
      
    </div>
    <dl class="transfer-details">
      <div>
        <dt>{props? props.category : "other"}</dt>
        <dd>{props? props.category : "other"}</dd>
      </div>
      <div>
        <dt>{props? props.date : ""}</dt>
        <dd>Date payment</dd>
      </div>
    </dl>
    <div class="transfer-number">
      {props? props.amount : 0}
    </div>
  </div>
)

const reverseTransactions = () => {
  displayTransactions(displayTransactions.reverse);
}

const displayTransactions = (bool) => {
  if (bool===true) {
    displayTransactions.reverse = false;
  }
  if (bool===false){
    displayTransactions.reverse = true;
  }
}

```

```

    }
    let val = document.querySelector("#balance")
    fetch("http://localhost:5000/api/expenses")
    .then(body=>body.json())
    .then((x)=>{
        let node = document.querySelector(".transfer-section")
        if (displayTransactions.reverse){
            x = x.reverse();
        }
        node
        .replaceChild(
            <div class="transfers">
                {x.map(row=>(<Transaction category={row.category}
date={row.date} amount={row.amount}/>))}
            </div>
            ,node.querySelector(".transfers"))
        })
    }

const delParent = (e) => {
    e.currentTarget.parentElement.remove();
}

const updateBalance = (e) => {
    e.preventDefault()
    let form = event.currentTarget;
    let form_data = new FormData();
    form_data.append("category", "others")
    let yourDate = new Date()
    form_data.append("date", yourDate.toISOString().split('T')[0])
    fetch("http://localhost:5000/api/balance")
    .then(body=>body.json())
    .then((x)=>{
        form_data.append("amount", +(x.balance)-
form.querySelector("input[name='amount']").value)
        fetch("http://localhost:5000/api/expenses", {
            method: "POST",
            body: form_data
        })
        .then(body=>body.json())
        .then((x)=>{
            if (!x.Success){
                //alert(x.msg);
                main();
            } else {
                window.location = "http://localhost:5000/dashboard"
            }
        })
    })
}

```

```

    })
  }

const addExpense = (e) => {
  e.preventDefault()
  let form = event.currentTarget;
  fetch("http://localhost:5000/api/expenses", {
    method: "POST",
    body: new FormData(form)
  })
  .then(body=>body.json())
  .then((x)=>{
    if (!x.Success){
      //alert(x.msg);
      main();
    } else {
      window.location = "http://localhost:5000/dashboard"
    }
  })
}

const openBalPopup = (e) => {
  var modal = document.getElementById("myModal");
  var btn = document.getElementById("myBtn");
  var span = document.getElementsByClassName("close")[0];
  btn.onclick = function() {
    modal.style.display = "block";
    document.querySelector("body").style.overflow="hidden";
  }
  span.onclick = function() {
    modal.style.display = "none";
    document.querySelector("body").style.overflow="unset";
  }
}

const logout = () => {
  fetch("http://localhost:5000/api/session",{
    method: "DELETE"
  })
  .then(()=>{
    window.location = "http://localhost:5000/"
  })
}

const main = () => {
  displayBalance()
  displayTransactions.reverse = true;
}

```

```

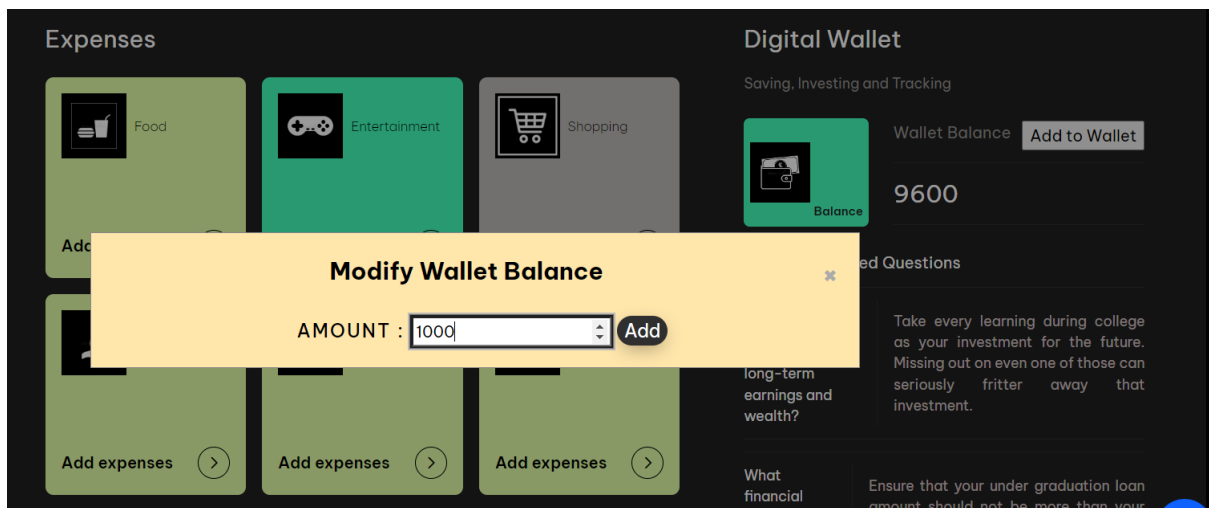
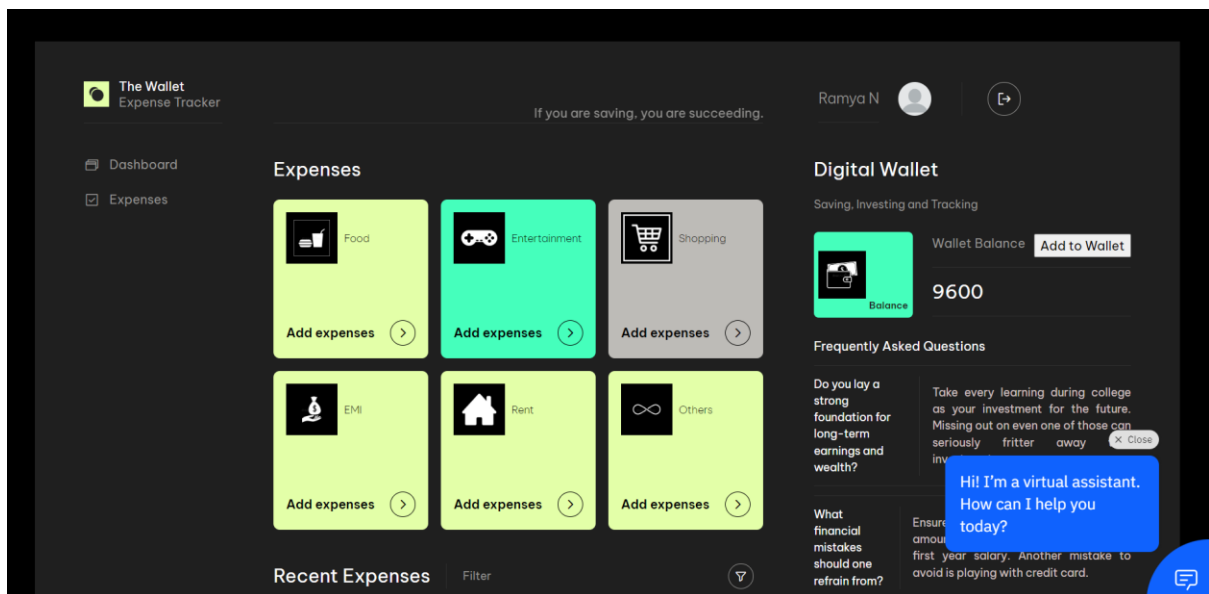
    displayTransactions()
    displayName()
}

window.onload = main

main()

```

Output:



ing, y

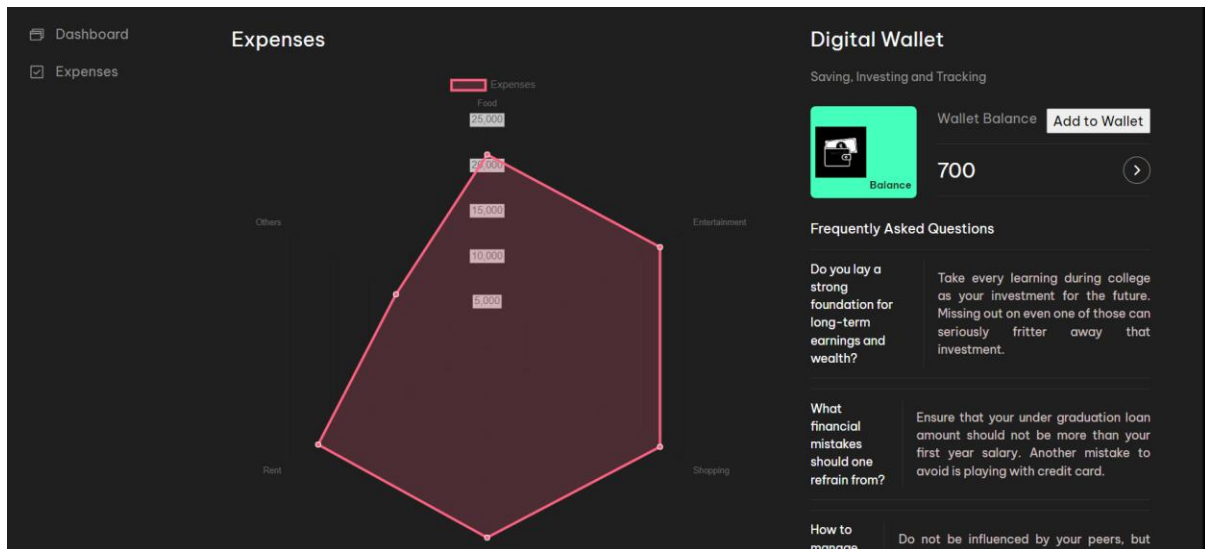
should one
refrain from?

first year salary. Another mistake to avoid is playing with credit card.

How to manage debt and boost your credit score?

Do not be influenced by your peers, but stick to maintaining your own budget. Take up any campus employment to balance your expenditures and make a saving, along with offering and gaining valued experience.

Graphical Representation of Expenses:



- Integrate Sendgrid and send emails using Sendgrid

Source Code:

```

166 def send(user, email):
167     print(email)
168     sql_stmt="SELECT SUM(amount) as balance from expenses where expenses.user=?"
169     stmt=ibm_db.prepare(conn,sql_stmt)
170     ibm_db.bind_param(stmt,1,user)
171     ibm_db.execute(stmt)
172     row = ibm_db.fetch_assoc(stmt)
173     ret = {}
174     if row != False and row["BALANCE"]!=None and row["BALANCE"]>0:
175         message=Mail(from_email="ramyanmsr@gmail.com",
176                     to_email=email,
177                     subject='ALERT!! Budget Exceeded!',
178                     plain_text_content='Budget has been crossed for this month',
179                     html_content='<strong>Hi user,<br>You have crossed your budget for thi
180
181         try:
182             sg = SendGridAPIClient(os.environ.get('SENDGRID_API_KEY'))
183             print(os.environ.get('SENDGRID_API_KEY'))
184             response=sg.send(message)
185             print(response.status_code)
186             print(response.body)
187             print(response.headers)
188         except Exception as e:
189             print(e)
190
191 if __name__ == "__main__":
192     app.run(host='0.0.0.0', debug=True)

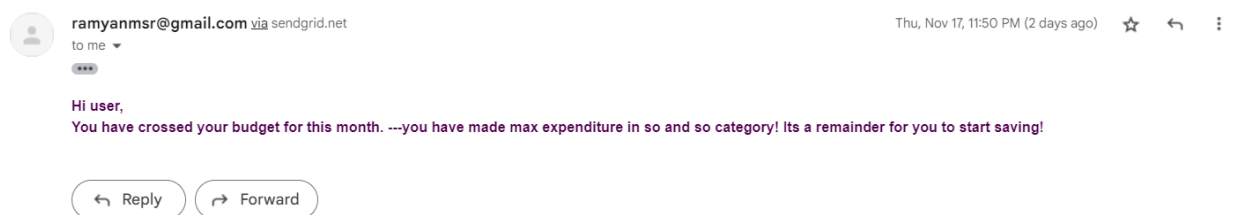
```

If expenditure exceeds the budget, a notification will be sent as an email:

```

ramyanmsr@gmail.com
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
127.0.0.1 - - [19/Nov/2022 14:22:33] "GET /api/balance HTTP/1.1" 200 -
202
b''
Server: nginx
Date: Sat, 19 Nov 2022 08:52:43 GMT
Content-Length: 0
Connection: close
X-Message-Id: udinYWVjRz2ztHEtAaMCdA
Access-Control-Allow-Origin: https://sendgrid.api-docs.io
Access-Control-Allow-Methods: POST
Access-Control-Allow-Headers: Authorization, Content-Type, On-behalf-of, x-sg-elas-acl
Access-Control-Max-Age: 600
X-No-CORS-Reason: https://sendgrid.com/docs/Classroom/Basics/API/cors.html
Strict-Transport-Security: max-age=600; includeSubDomains

```



- **Integrating Chatbot (IBM Watson) with the Application**

