ESD (Employability Skills Development) FA-3

Title:

Call the API through JavaScript and show it in html. Also implement all CRUD operations - Expense Tracker App

Tools: Microsoft Word, PhpMyAdmin, Postman

Screenshot:

Database Design

	Table	Action	Rows	Type	Collation	Size	Overhead
☐	customer	Browse Structure Search Insert Empty Drop	4	InnoDB	utf8mb4_general_ci	16.0 KiB	-
☐	expense	Browse Structure Search Insert Empty Drop	1	InnoDB	utf8mb4_general_ci	48.0 KiB	-
☐	payment	Browse Structure Search Insert Empty Drop	1	InnoDB	utf8mb4_general_ci	16.0 KiB	-
	3 tables	Sum	6	InnoDB	utf8mb4_general_ci	80.0 KiB	0 B

Structure of Customer Table

Server: 127.0.0.1:3306 » Database: esd_grp. 13 » Table: customer

Browse | Structure | SQL | Search | Insert | Export | Import | Privileges | Operation

Table structure | Relation view

	#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
☐	1	id	int(100)			No	*None*			Change Drop More
☐	2	name	varchar(700)	utf8mb4_general_ci		No	*None*			Change Drop More
☐	3	email	varchar(255)	utf8mb4_general_ci		No	*None*			Change Drop More
☐	4	payment	varchar(255)	utf8mb4_general_ci		No	*None*			Change Drop More
☐	5	phone	varchar(255)	utf8mb4_general_ci		No	*None*			Change Drop More

Table: Customer

Showing rows 0 - 3 (4 total, Query took 0.0023 seconds.)

```
SELECT * FROM `customer`
```

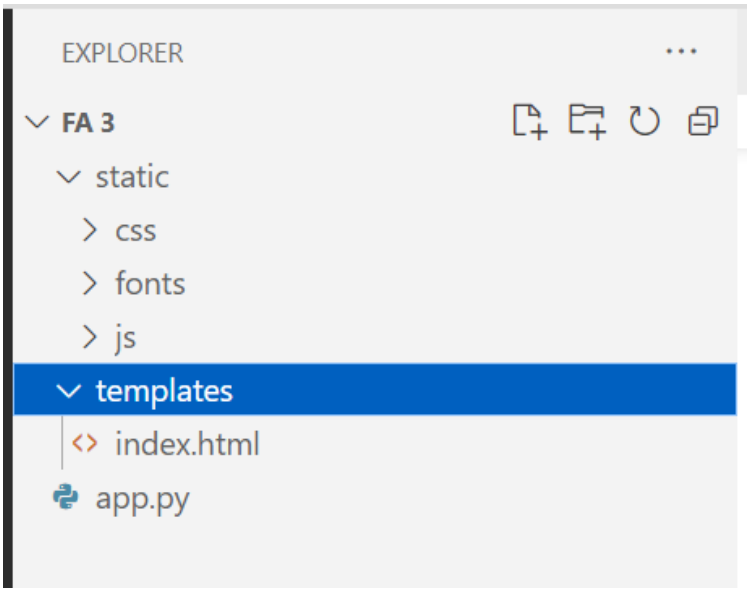
Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

Show all | Number of rows: 25 Filter rows: Search this table Sort by key: None

+ Options

	id	name	email	payment	phone
Edit Copy Delete	10	Samar Raichand	samar2005@gmail.com	10000	9892831863
Edit Copy Delete	11	Arya Tiwari	aryatiwari2003@gmail.com	11000	9320408279
Edit Copy Delete	12	Kinjal Singh	kinjal2003@gmail.com	12000	7678073799
Edit Copy Delete	13	Divyanshu Singh	divyanshu2003@gmail.com	13000	7738296424

Project Directory

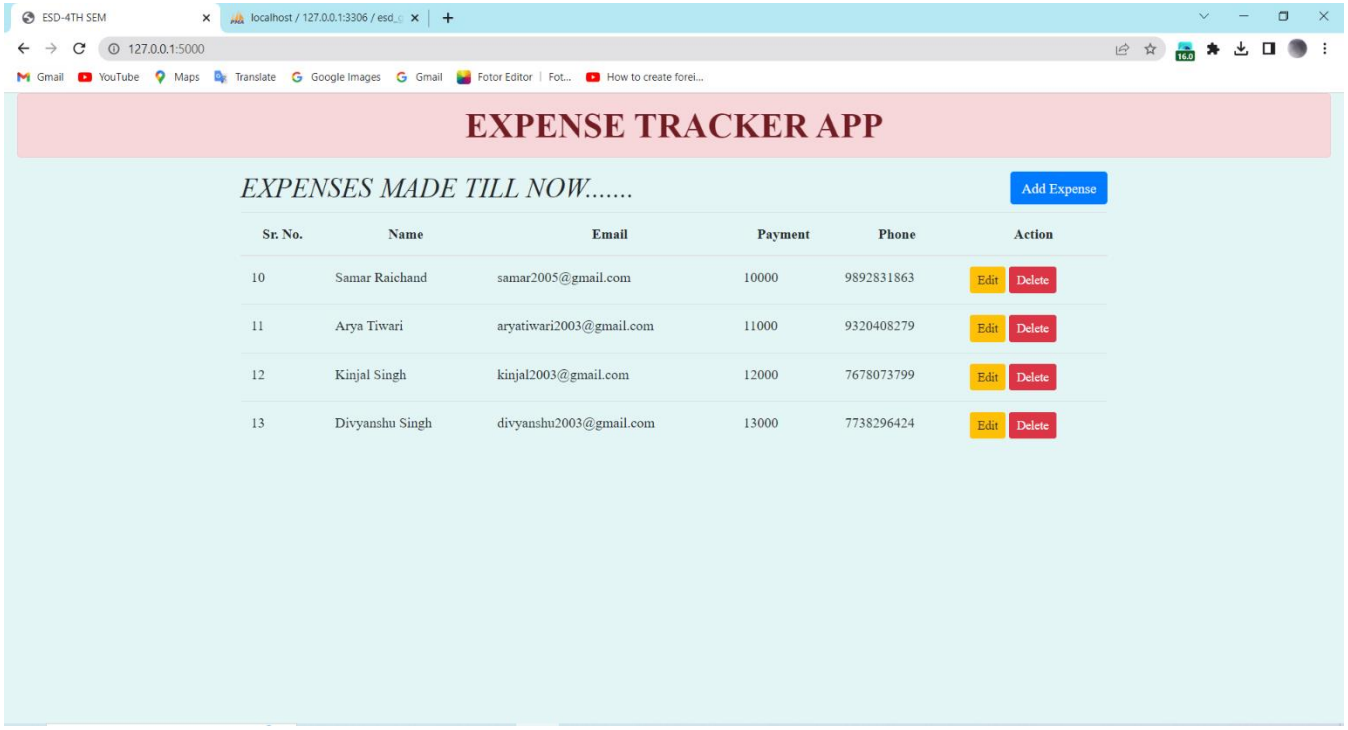


app.py

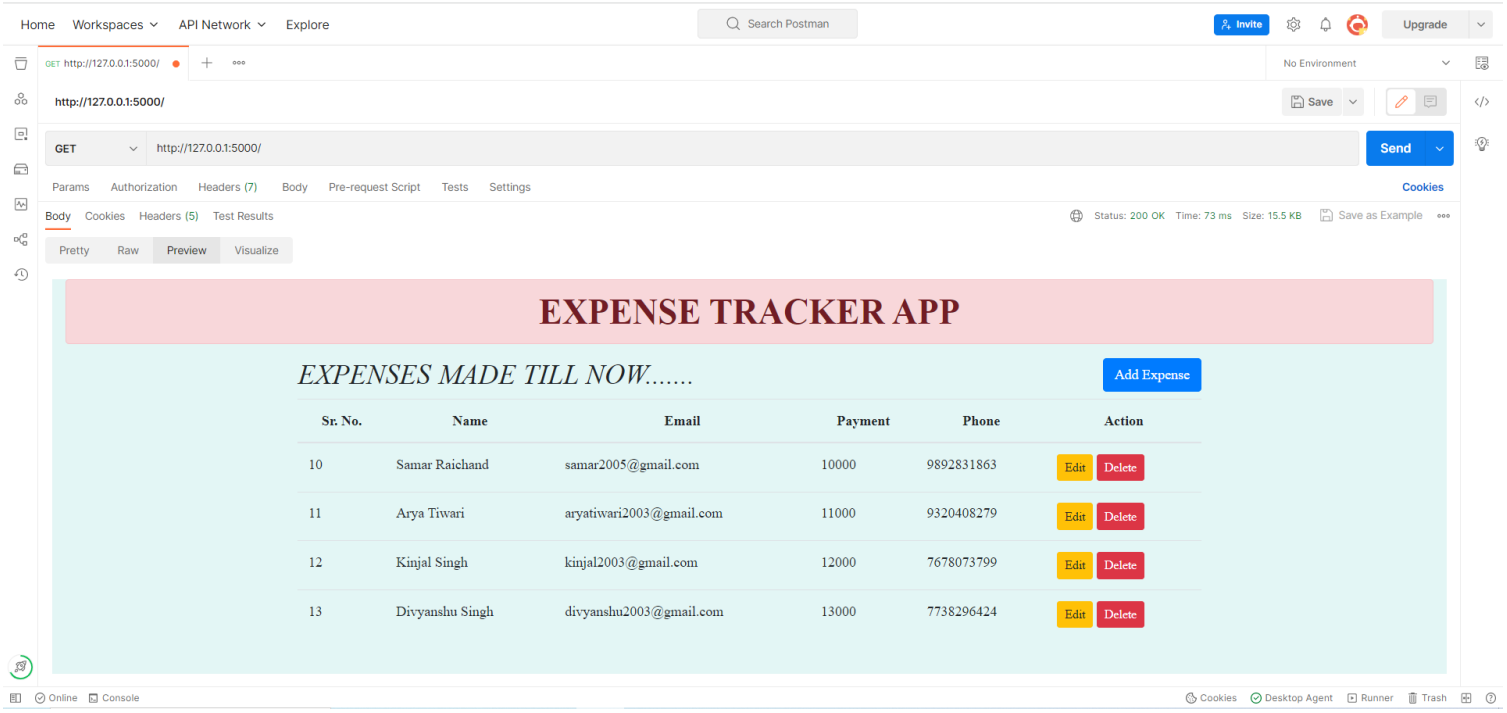
```
from flask import Flask,
render_template, request, url_for, flash
from werkzeug.utils import redirect
from flask_mysqldb import MySQL
app = Flask(__name__)
app.secret_key = ''

app.config['MYSQL_HOST'] =
'localhost'
app.config['MYSQL_USER'] = 'root'
```

```
app.config['MYSQL_PASSWORD'] = ''
app.config['MYSQL_DB'] = 'esd_grp.
13'

mysql = MySQL(app)

@app.route('/')

def Index():
    cur = mysql.connection.cursor()
    cur.execute("SELECT * FROM
customer")
    data = cur.fetchall()
    cur.close()

    return render_template('index.html',
customer=data)


@app.route('/insert', methods = ['POST'])
def insert():

  if request.method == "POST":
        flash("Data Inserted Successfully")
name = request.form.get('name')
        email = request.form.get('email')
        payment =
request.form.get('payment')
        phone = request.form.get('phone')
        cur = mysql.connection.cursor()

 cur.execute("INSERT INTO customer
(name, email, payment, phone)
VALUES (%s, %s, %s, %s)", (name,
email, payment, phone))
        mysql.connection.commit()

return redirect(url_for('Index'))


@app.route('/delete/<string:id_data>',
methods = ['GET'])

def delete(id_data):
    flash("Record Has Been Deleted
Successfully")

cur = mysql.connection.cursor()
    cur.execute("DELETE FROM
customer WHERE id=%s", (id_data,))
    mysql.connection.commit()
    return redirect(url_for('Index'))


@app.route('/update',
methods= ['POST', 'GET'])
def update():

 if request.method == 'POST':
        id = request.form.get('id')
        name = request.form.get('name')
        email = request.form.get('email')
        payment =
request.form.get('payment')
        phone = request.form.get('phone')
        cur = mysql.connection.cursor()

 cur.execute("""
        UPDATE customer SET name=%s,
email=%s, payment=%s, phone=%s
        WHERE id=%s
        """, (name, email, payment, phone,
id_data))
        flash("Data Updated Successfully")
        return redirect(url_for('Index'))

if __name__ == "__main__":
    app.run(debug=True)
```

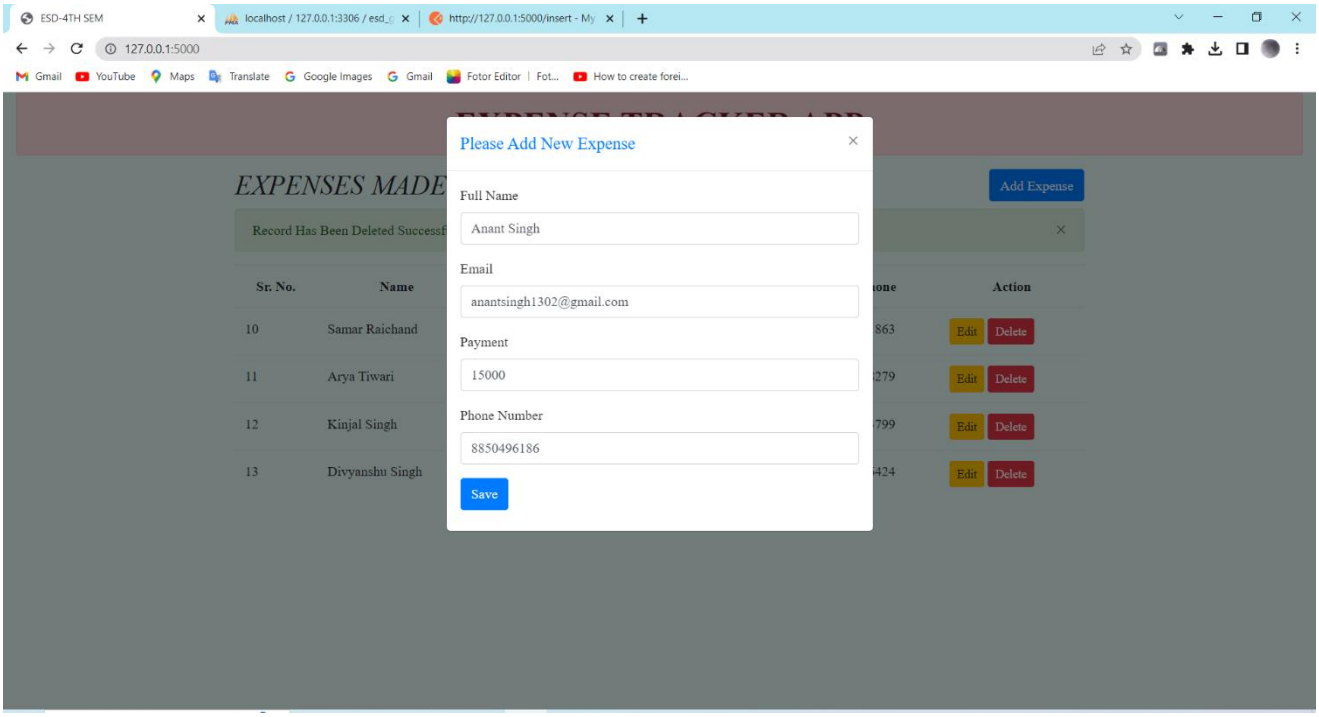
MAIN WEB PAGE

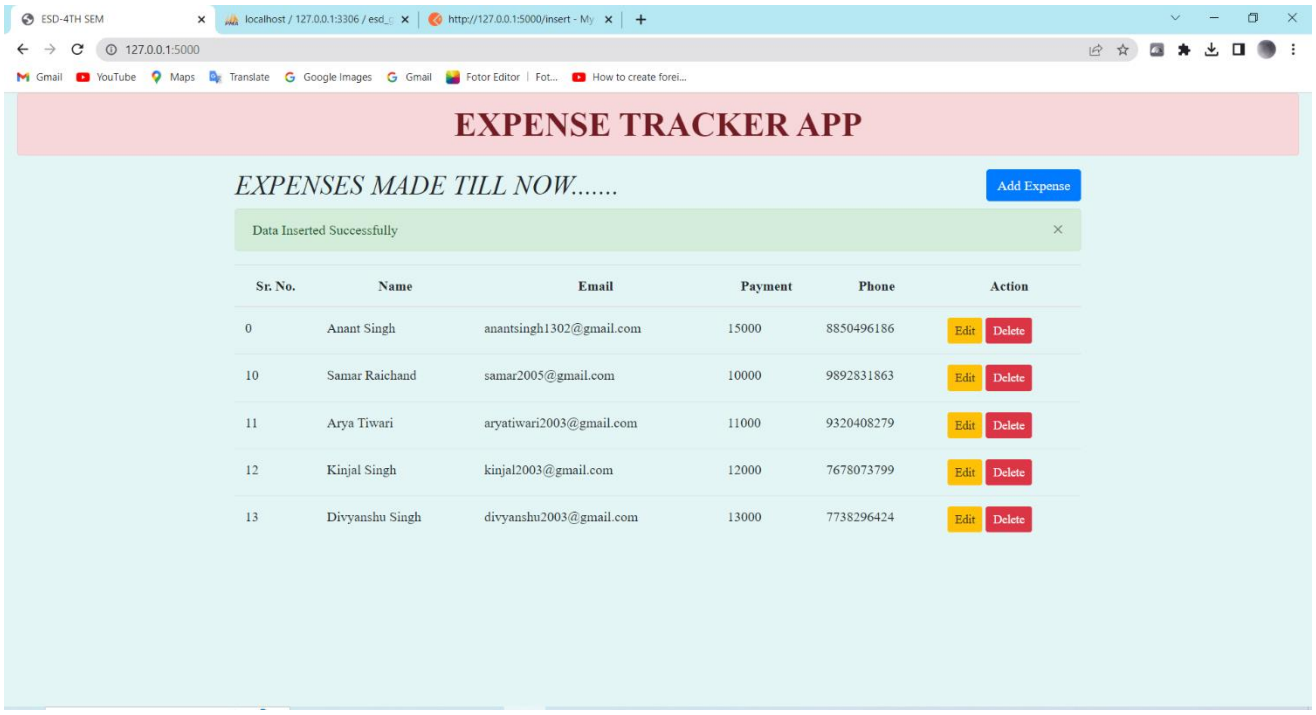


SHOWING THE EXPENSES MADE BY THE CUSTOMER THROUGH GET METHOD

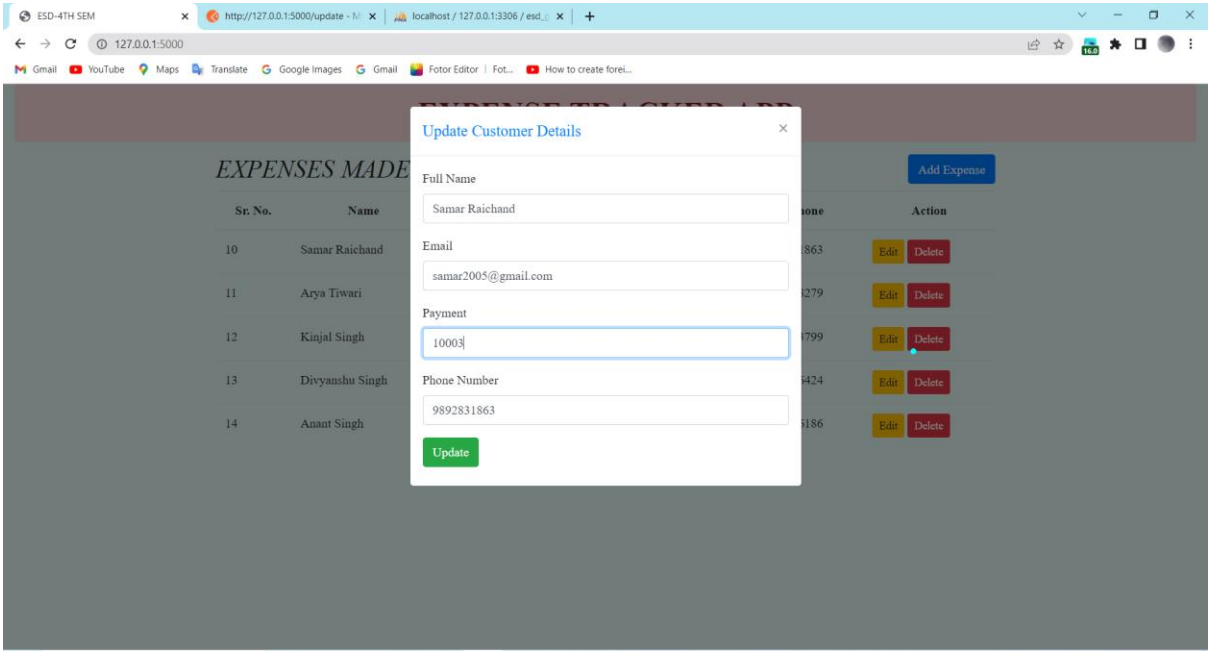


ADDING THE NEW EXPENSES

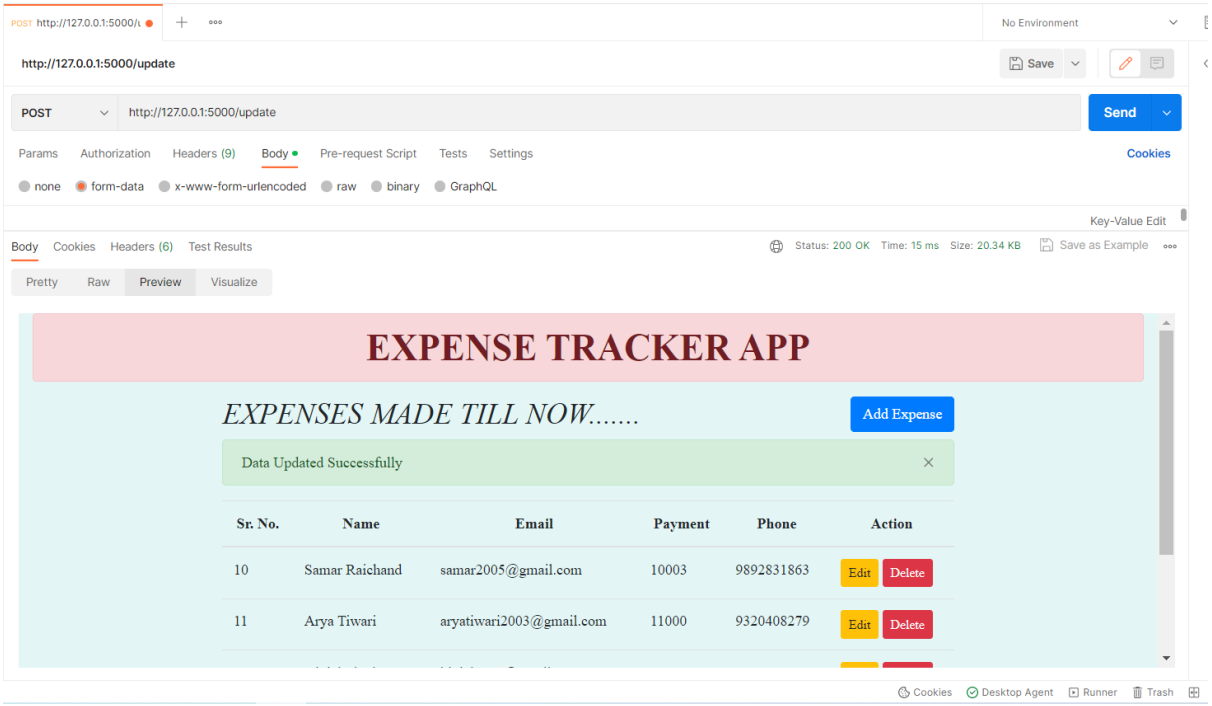




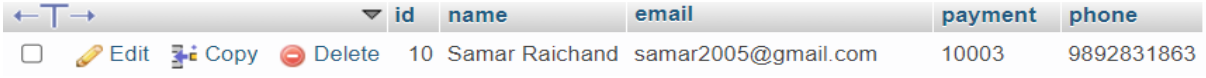
UPDATING THE EXPENSES



Entering the details to update

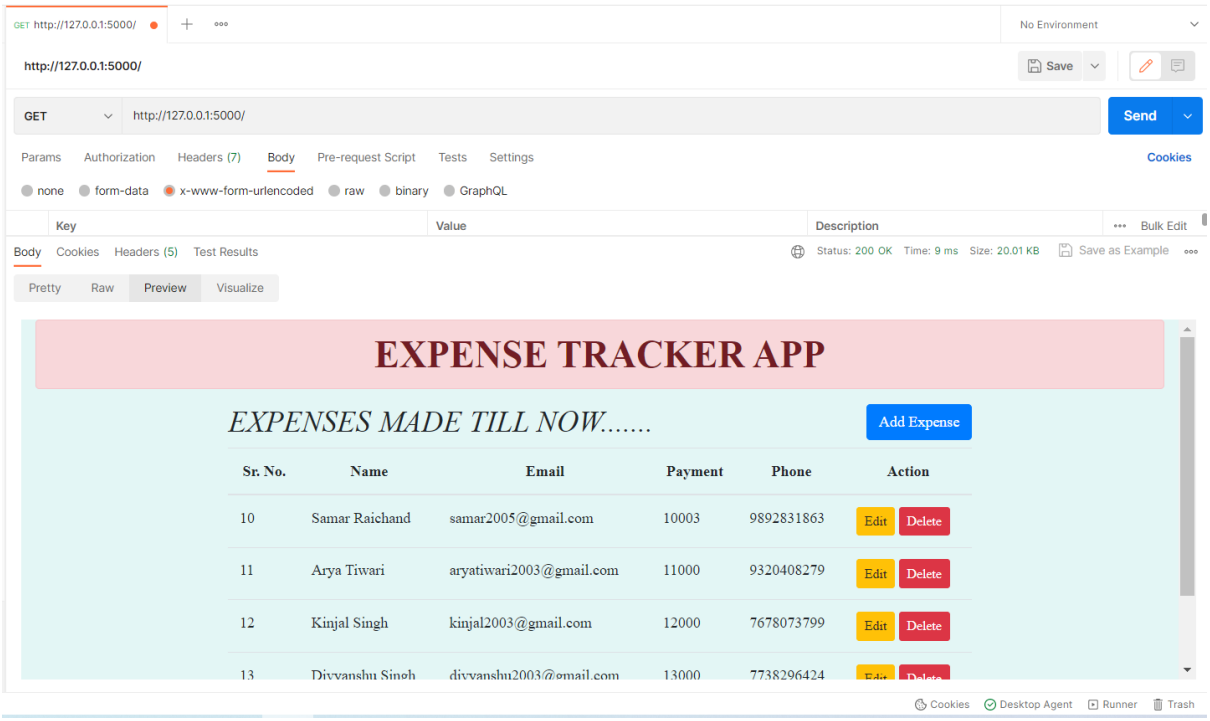


Updating a record by calling the API

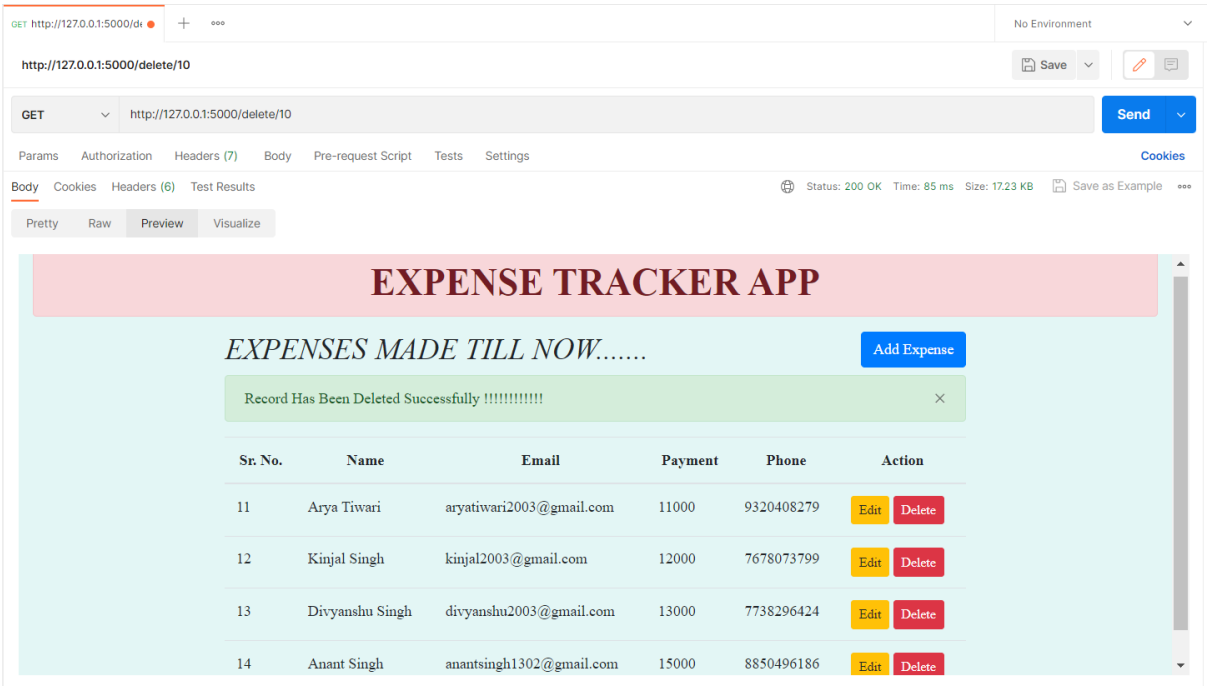


Database after performing the CRUD operation i.e. Update

DELETING THE EXPENSES



Before deleting a record by calling the API



After deleting a record by calling the API

Showing rows 0 - 3 (4 total, Query took 0.0010 seconds.)

```
SELECT * FROM `customer`
```

Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

Show all | Number of rows: 25 Filter rows: Search this table Sort by key: None

+ Options

	id	name	email	payment	phone
Edit Copy Delete	11	Arya Tiwari	aryatiwari2003@gmail.com	11000	9320408279
Edit Copy Delete	12	Kinjal Singh	kinjal2003@gmail.com	12000	7678073799
Edit Copy Delete	13	Divyanshu Singh	divyanshu2003@gmail.com	13000	7738296424
Edit Copy Delete	14	Anant Singh	anantsingh1302@gmail.com	15000	8850496186

Check all With selected: Edit Copy Delete Export

Database after performing the CRUD operation i.e. Delete