

CSIT 555  
DATABASE SYSTEMS  
FINAL PROJECT – TAX TRACKER SYSTEM  
BY  
SQL SULTANS

SAI ROHINI GODAVARTHI  
RAHUL EGA  
HEMANTH KUMAR BHUVANAGIRI  
NAVEEN REDDY BOKKA  
PRANAY THARALA

### Tax Tracker Company:

Tax Tracker Company streamlines managing and tracking tax-related data for companies and customers. The application provides a secure login system for employees and customers, allowing them to interact with and manage relevant information.

GitHub link: [https://github.com/ro468/DBMS\\_Project.git](https://github.com/ro468/DBMS_Project.git)

YouTube link: <https://youtu.be/hJqs9gPCMIs>

Following are the screenshots of the code (.py and .html):

#### app.py

```
 1  from flask import Flask, render_template, request, g, redirect
 2  import sqlite3
 3  from datetime import datetime
 4
 5  app = Flask(__name__)
 6
 7  # Define the database name
 8  DATABASE = 'my_database.db' # Replace 'your_database_name' with the actual database name
 9
10
11 # Function to connect to the database
12 usages
12 def get_db():
13     db = getattr(g, '_database', None)
14     if db is None:
15         db = g._database = sqlite3.connect(DATABASE)
16     return db
17
18
19 # Function to close the database connection
20 @app.teardown_appcontext
21 def close_connection(exception):
22     db = getattr(g, '_database', None)
23     if db is not None:
24         db.close()
25
26
27 # Define the route for the home page
28 @app.route('/')
29 def home():
30     return render_template('index.html')
```

12/5/2023

```
33     # Define the route for employee login
34     @app.route(rule: '/employee_login', methods=['GET', 'POST'])
35     def employee_login():
36         if request.method == 'POST':
37             password = request.form['password']
38
39             # For simplicity, use a fixed password '123' for all employees
40             if password == '123':
41                 # Fetch all companies from the database
42                 cursor = get_db().cursor()
43                 cursor.execute('SELECT DISTINCT category FROM my_table')
44                 companies = cursor.fetchall()
45                 return render_template(template_name_or_list: 'employee_dashboard.html', companies=companies)
46             else:
47                 return "Invalid password. Please try again."
48
49     return render_template('employee_login.html') # Display the login form for GET requests
50
51
52
53     # Define the route for customer login
54
55     @app.route(rule: '/customer_login', methods=['GET', 'POST'])
56     def customer_login():
57         if request.method == 'POST':
58             customer_category = request.form['customer_category']
59
60             # Fetch customer information from the database
61             cursor = get_db().cursor()
62             cursor.execute(_sql: 'SELECT category FROM my_table WHERE category = ?', _parameters: (customer_category,))
63             category = cursor.fetchone()
64
65             if category:
66                 # Redirect to the appropriate category route
67                 return redirect(f'/company_info/{category[0]}')
68             else:
69                 return "Customer not found. Please check the category and try again."
70
71     return render_template('customer_login.html') # Display the login form for GET requests
72
73
74     # Define the route for new user signup
75     @app.route(rule: '/new_user_signup', methods=['GET', 'POST'])
76     def new_user_signup():
77         if request.method == 'POST':
78             # Generate a unique consecutive ID (assuming it's the next available ID in the database)
79             cursor = get_db().cursor()
80             cursor.execute('SELECT MAX(id) FROM my_table')
81             last_id = cursor.fetchone()[0]
82             new_id = last_id + 1 if last_id is not None else 1
83
84             # Get other form data
85             company = request.form['company']
86             amount = request.form['amount']
87
88             # Pass the current year to the template
89             current_year = datetime.now().year
90
91             # Insert new user data into the database
92             cursor.execute(_sql: 'INSERT INTO my_table VALUES (?, ?, ?, ?, ?, ?, ?)', _parameters: (new_id, company, amount, None, 'unpaid', f'{current_year}-01-15'))
```

12/5/2023

```
95     # Commit the changes to the database
96     get_db().commit()
97
98     return f"New user with ID {new_id} successfully created!", 200, {'Content-Type': 'text/plain'}
99
100    return render_template(template_name_or_list: 'new_user_signup.html', current_year=datetime.now().year)
101
102 @app.route('/company_info/<company>')
103 def company_info(company):
104     print("Company:", company) # Add this line for debugging
105     # Fetch company information from the database
106     cursor = get_db().cursor()
107     cursor.execute(_sql: 'SELECT * FROM my_table WHERE category = ?', __parameters: (company,))
108     company_info = cursor.fetchall()
109     print("Company Info:", company_info) # Add this line for debugging
110     return render_template(template_name_or_list: 'company_info.html', company_info=company_info, company=company)
111
112
113 # Define the route for paying the invoice
114 @app.route(rule: '/pay_here', methods=['POST'])
115 def pay_here():
116     payment_id = request.form['payment_id']
117
118     # Update the payment status in the database
119     cursor = get_db().cursor()
120     cursor.execute(_sql: 'UPDATE my_table SET status = "paid" WHERE id = ?', __parameters: (payment_id,))
121     get_db().commit()
122
123     return "You've successfully paid. Thanks for being an amazing Taxpayer!"
124
125
126 # Edit record
127 @app.route(rule: '/edit_record/<record_id>', methods=['GET', 'POST'])
128 def edit_record(record_id):
129     success_message = None
130
131     if request.method == 'POST':
132         new_amount = request.form['new_amount']
133         due_date_str = request.form['due_date']
134         month, day = due_date_str.split()
135         month_number = datetime.strptime(month, __format: "%B").month
136         current_year = datetime.now().year
137         next_year = current_year + 1
138         due_date = datetime(next_year, month_number, int(day))
139
140         cursor = get_db().cursor()
141         cursor.execute(_sql: 'UPDATE my_table SET amount = ?, due_date = ? WHERE id = ?', __parameters: (new_amount, due_date, record_id))
142         get_db().commit()
143         success_message = 'Changes successfully submitted!'
144
145     # Fetch the company category for the edited record
146     cursor.execute(_sql: 'SELECT category FROM my_table WHERE id = ?', __parameters: (record_id,))
147     category = cursor.fetchone()[0]
148
149     return redirect(f'/company_info/{category}?success_message={success_message}')
150
151
152     cursor = get_db().cursor()
153     cursor.execute(_sql: 'SELECT * FROM my_table WHERE id = ?', __parameters: (record_id,))
154     record = cursor.fetchone()
155     return render_template(template_name_or_list: 'edit_record.html', record=record, success_message=success_message)
```

12/5/2023

```
157     # Delete record
158     @app.route('/delete_record/<record_id>')
159     def delete_record(record_id):
160         cursor = get_db().cursor()
161         cursor.execute(_sql: 'DELETE FROM my_table WHERE id = ?', __parameters: (record_id,))
162         get_db().commit()
163         return redirect('/')
164
165     # Save record
166     @app.route(rule: '/save_record', methods=['POST'])
167     def save_record():
168         record_id = request.form['record_id']
169         new_amount = request.form['new_amount']
170         cursor = get_db().cursor()
171         cursor.execute(_sql: 'UPDATE my_table SET amount = ? WHERE id = ?', __parameters: (new_amount, record_id))
172         get_db().commit()
173         return redirect('/')
174
175 ▷ if __name__ == '__main__':
176     app.run(debug=True)
177
178
```

## db.py

```
1 import sqlite3
2
3 # Connect to SQLite database (this will create a new database if it doesn't exist)
4 connection = sqlite3.connect('my_database.db')
5
6 # Create a cursor object to execute SQL queries
7 import sqlite3
8
9 # Connect to SQLite database (this will create a new database if it doesn't exist)
10 connection = sqlite3.connect('my_database.db') # Replace 'your_database_name' with the desired name
11
12 # Create a cursor object to execute SQL queries
13 cursor = connection.cursor()
14
15 # Define the table schema
16 create_table_query = """
17 CREATE TABLE IF NOT EXISTS my_table (
18     id INTEGER PRIMARY KEY,
19     category TEXT,
20     amount REAL,
21     date_paid TEXT,
22     status TEXT,
23     due_date TEXT
24 );
25 """
26
27 # Execute the create table query
28 cursor.execute(create_table_query)
```

12/5/2023

```
27 # Execute the create table query
28 cursor.execute(create_table_query)
29
30 # Insert data into the table
31 data = [
32     (1, 'derm', 4100.00, '2023-09-26', 'paid', '2024-01-15'),
33     (2, 'derm', 4100.00, '2023-10-12', 'paid', '2024-01-15'),
34     (3, 'tek', 15200.00, '2023-06-09', 'paid', '2023-06-15'),
35     (4, 'tek', 15200.00, '2023-07-12', 'paid', '2023-09-15'),
36     (5, 'tek', 11400.00, '2023-08-11', 'paid', '2023-09-15'),
37     (6, 'tek', 14440.00, '2023-09-21', 'paid', '2024-01-15'),
38     (7, 'tek', 15200.00, '2023-10-18', 'paid', '2024-01-15'),
39     (8, 'tek', 23520.00, None, 'unpaid', '2024-01-15'), # Use None for NULL
40     (9, 'tek', 16800.00, None, 'unpaid', '2024-01-15'),
41     (10, 'tek', 16800.00, None, 'unpaid', '2024-01-15'),
42     (11, 'tek', 16800.00, None, 'unpaid', '2024-01-15')
43 ]
44
45 # Insert data into the table
46 cursor.executemany(_sql: 'INSERT INTO my_table VALUES (?, ?, ?, ?, ?, ?)', data)
47
48 # Commit the changes and close the connection
49 connection.commit()
50 connection.close()
51
52
53
```

## Index.html

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4     <meta charset="UTF-8">
5     <meta name="viewport" content="width=device-width, initial-scale=1.0">
6     <title>TAX TRACKER COMPANY</title>
7     <link rel="stylesheet" href="https://fonts.googleapis.com/css2?family=Montserrat:wght@400;700&display=swap">
8     <style>
9         body {
10             font-family: 'Montserrat', sans-serif;
11             background: linear-gradient(135deg, #ff6b6b, #6b6b6b);
12             margin: 0;
13             padding: 0;
14             display: flex;
15             align-items: center;
16             justify-content: center;
17             height: 100vh;
18         }
19
20         .container {
21             text-align: center;
22             background-color: rgba(255, 255, 255, 0.9);
23             padding: 40px;
24             border-radius: 8px;
25             box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
26             width: 80%;
27             max-width: 800px;
28         }
29
30         h1 {
31             color: #333;
32             background-color: #3498db;
33             padding: 10px;
```

12/5/2023

```
34     border-radius: 8px;
35     margin-bottom: 20px;
36   }
37
38   h2 {
39     text-transform: uppercase;
40     letter-spacing: 2px;
41     color: black;
42     margin-bottom: 20px;
43   }
44
45   .card-container {
46     display: flex;
47     justify-content: space-around;
48     flex-wrap: wrap;
49     margin-top: 20px;
50   }
51
52   .card {
53     flex: 1;
54     margin: 10px;
55     background-color: #3498db;
56     color: #fff;
57     padding: 20px;
58     border-radius: 8px;
59     box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
60     transition: background-color 0.3s ease, color 0.3s ease;
61   }
62
63   .card:hover {
64     background-color: #217dbb;
65   }
66
67   form {
68     display: flex;
69     flex-direction: column;
70     align-items: center;
71   }
72
73   label {
74     display: block;
75     margin-bottom: 8px;
76     color: #fff;
77   }
78
79   input {
80     width: 100%;
81     padding: 10px;
82     margin-bottom: 16px;
83     border: 1px solid #ccc;
84     border-radius: 4px;
85     box-sizing: border-box;
86   }
87
88   input[type="submit"] {
89     background-color: #fff;
90     color: #3498db;
91     padding: 12px;
92     border: none;
93     border-radius: 4px;
94     cursor: pointer;
95     font-size: 16px;
96     transition: background-color 0.3s ease, color 0.3s ease;
97   }
98 }
```

12/5/2023

```
99         input[type="submit"]:hover {
100             background-color: #eee;
101         }
102     </style>
103 </head>
104 <body>
105
106     <div class="container">
107         <h1>TAX TRACKER COMPANY</h1>
108
109         <div class="card-container">
110             <div class="card">
111                 <h2>EMPLOYEE LOGIN</h2>
112                 <form action="/employee_login" method="post">
113                     <label for="employee-password">Password:</label>
114                     <input type="password" id="employee-password" name="password" placeholder="Enter your password">
115                     <input type="submit" value="Login">
116                 </form>
117             </div>
118
119             <div class="card">
120                 <h2>CUSTOMER LOGIN</h2>
121                 <form action="/customer_login" method="post">
122                     <label for="customer-category">Category:</label>
123                     <input type="text" id="customer-category" name="customer_category" placeholder="Enter your category">
124                     <input type="submit" value="Login">
125                 </form>
126             </div>
127         </div>
128
129         <h2>New User Signup</h2>
130         <form action="/new_user_signup">
131             <input type="submit" value="Signup">
132         </form>
133     </div>
134
135 </body>
136 </html>
137
```

12/5/2023

## employee\_login.html

```
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <meta name="viewport" content="width=device-width, initial-scale=1.0">
6      <title>Employee Login</title>
7  </head>
8  <body>
9      <h1>Employee Login</h1>
10
11     <form action="/employee_login" method="post">
12         Password: <input type="password" name="password" id="password" placeholder="Enter password">
13         <input type="submit" value="Login">
14
15         <script>
16             document.addEventListener('DOMContentLoaded', function () {
17                 // JavaScript to handle password masking
18                 var passwordInput = document.getElementById('password');
19
20                 passwordInput.addEventListener('input', function () {
21                     // Display dots ('...') as the user types
22                     this.value = '...'.substring(0, this.value.length);
23                 });
24
25                 passwordInput.addEventListener('focus', function () {
26                     // Clear the input field when it is focused
27                     this.value = '';
28                 });
29
30                 passwordInput.addEventListener('focus', function () {
31                     // Clear the input field when it is focused
32                     this.value = '';
33                 });
34
35                 passwordInput.addEventListener('blur', function () {
36                     // If the input is left empty, display a placeholder
37                     if (this.value === '') {
38                         this.value = 'Enter password';
39                     }
40                 });
41             });
42         </script>
43     </form>
44  </body>
45  </html>
```

12/5/2023

## customer\_login.html

```
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <meta name="viewport" content="width=device-width, initial-scale=1.0">
6      <title>Customer Login</title>
7  </head>
8  <body>
9      <h1>Customer Login</h1>
10
11     <form action="/customer_login" method="post">
12         Category: <input type="text" name="customer_category">
13         <input type="submit" value="Login">
14     </form>
15 </body>
16 </html>
```

## employee\_dashboard.html

```
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <meta name="viewport" content="width=device-width, initial-scale=1.0">
6      <title>Employee Dashboard</title>
7
8  <style>
9      body {
10          font-family: 'Segoe UI', Tahoma, Geneva, Verdana, sans-serif;
11          background: linear-gradient(to right, #ff5858, #f857a6, #8449e6, #434343);
12          color: #fff;
13          margin: 0;
14          display: flex;
15          align-items: center;
16          justify-content: center;
17          height: 100vh;
18      }
19
20      .dashboard-container {
21          text-align: center;
22          padding: 20px;
23          border-radius: 10px;
24          background-color: rgba(255, 255, 255, 0.8);
25          box-shadow: 0 0 20px rgba(0, 0, 0, 0.2);
26          max-width: 600px;
27          width: 100%;
28      }
29  </style>
```

12/5/2023

```
30     h1 {
31         font-family: 'Bradley Hand', cursive; /* Use Bradley Hand cursive font */
32         color: #fff;
33         background: linear-gradient(to right, #3498db, #5a5a5a);
34         padding: 10px;
35         border-radius: 5px;
36         margin-bottom: 20px;
37     }
38
39     h2 {
40         color: #0056b3;
41     }
42
43     .company-list {
44         list-style-type: none;
45         padding: 0;
46         margin-top: 20px;
47     }
48
49     .company-list li {
50         margin-bottom: 10px;
51         background: linear-gradient(to right, #3498db, #5a5a5a);
52         padding: 10px;
53         border-radius: 5px;
54     }
55
56     a {
57         text-decoration: none;
58         color: #fff;
59         font-weight: bold;
60     }
61
62
63         a:hover {
64             text-decoration: underline;
65         }
66     
```

```
</style>
67 </head>
68 <body>
69     <div class="dashboard-container">
70         <h1>Welcome to the Employee Dashboard</h1>
71
72         <h2>Companies:</h2>
73         <ul class="company-list">
74             {% for company in companies %}
75                 <li><a href="{{ url_for('company_info', company=company[0]) }}>{{ company[0] }}</a></li>
76             {% endfor %}
77         </ul>
78     </div>
79 </body>
80 </html>
```

**customer\_dashboard.html**

```
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <meta name="viewport" content="width=device-width, initial-scale=1.0">
6      <title>Customer Dashboard</title>
7  </head>
8  <body>
9      <h1>Welcome to the Customer Dashboard</h1>
10
11     <h2>Customer Information:</h2>
12     <table border="1">
13         <tr>
14             <th>ID</th>
15             <th>Category</th>
16             <th>Amount</th>
17             <th>Date Paid</th>
18             <th>Status</th>
19             <th>Due Date</th>
20         </tr>
21         {% for row in customer_info %}
22             <tr>
23                 <td>{{ row[0] }}</td>
24                 <td>{{ row[1] }}</td>
25                 <td>{{ row[2] }}</td>
26                 <td>{{ row[3] }}</td>
27                 <td>{{ row[4] }}</td>
28                 <td>{{ row[5] }}</td>
29             </tr>
30         {% endfor %}
31     </table>
32 </body>
33 </html>
```

## company\_info.html

```

1   <!DOCTYPE html>
2   <html lang="en">
3   <head>
4       <meta charset="UTF-8">
5       <meta name="viewport" content="width=device-width, initial-scale=1.0">
6       <title>Company Information</title>
7       <style>
8           body {
9               font-family: 'Arial', sans-serif;
10              background: linear-gradient(135deg, #4285f4, #0f9d58, #f4b400, #db4437);
11              color: #fff; /* White text */
12              margin: 0;
13              padding: 20px;
14              display: flex;
15              align-items: center;
16              justify-content: center;
17              height: 100vh;
18          }
19
20          h1 {
21              text-align: center;
22          }
23
24          .cards-container {
25              display: flex;
26              flex-wrap: wrap;
27              justify-content: center;
28          }
29
30          .company-info {
31              background-color: rgba(255, 255, 255, 0.9); /* White background with some transparency */
32              border-radius: 8px;
33
34              box-shadow: 0 0 10px rgba(0, 0, 0, 0.1); /* Slight shadow effect */
35              padding: 20px;
36              margin: 10px;
37              width: 300px;
38              text-align: left;
39              color: #000; /* Black text */
40              font-family: 'Cursive', sans-serif; /* Change font to cursive */
41          }
42
43          .company-info p {
44              margin: 0;
45              padding: 8px 0;
46          }
47
48          .company-info strong {
49              color: #4285f4; /* Google blue for strong elements */
50          }
51
52          .payment-button, .edit-button {
53              background-color: #4285f4; /* Google blue for buttons */
54              color: #fff; /* White text on buttons */
55              padding: 10px;
56              border: none;
57              border-radius: 4px;
58              cursor: pointer;
59              margin-top: 10px;
60          }
61
62          .payment-button:hover, .edit-button:hover {
63              background-color: #3367d6; /* Darker blue on hover */
64          }

```

```

65     hr {
66         border: 0;
67         border-top: 1px solid rgba(255, 255, 255, 0.7); /* Light gray line with some transparency */
68         margin: 20px 0;
69     }
70
71     p.no-info {
72         text-align: center;
73         color: #fff; /* White text for no information message */
74     }
75 
```

</style>

```

76 </head>
77 <body>
78     <h1>Company Information</h1>
79
80     <div class="cards-container">
81         {% if company_info %}
82             {% for row in company_info %}
83                 <div class="company-info">
84                     <p><strong>ID:</strong> {{ row[0] }}</p>
85                     <p><strong>Company:</strong> {{ row[1] }}</p>
86                     <p><strong>Amount:</strong> {{ row[2] }}</p>
87                     <p><strong>Status:</strong> {{ row[4] }}</p>
88
89                     {% if row[3] %}
90                         <!-- Display Date Paid if available -->
91                         <p><strong>Date Paid:</strong> {{ row[3] }}</p>
92                     {% endif %}
93
94                     {% if row[5] %}
95                         <!-- Display Due Date if available -->
96                         <p><strong>Due Date:</strong> {{ row[5] }}</p>
97                     {% endif %}
98
99
100                {% if row[4] == 'unpaid' %}
101                    <!-- Display the "Pay Here" button for unpaid status -->
102                    <form action="/pay_here" method="post">
103                        <input type="hidden" name="payment_id" value="{{ row[0] }}"/>
104                        <button class="payment-button" type="submit">Pay Here</button>
105                    {% endif %}
106
107
108                    <!-- Display the "Edit" button -->
109                    <button class="edit-button" onclick="editRow('{{ row[0] }}')>Edit</button>
110                </div>
111
112                {% if not loop.last %}
113                    <hr>
114                {% endif %}
115            {% endfor %}
116
117            {% else %}
118                <p class="no-info">No information available for this company.</p>
119            {% endif %}
120
121        </div>
122
123        <!-- JavaScript function to handle edit -->
124        <script>
125            function editRow(recordId) {
126                // Redirect to the edit page with the record ID
127                window.location.href = "/edit_record/" + recordId;
128            }
129        </script>
130
131    </body>
132
133 
```

## edit\_record.html

```
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <meta name="viewport" content="width=device-width, initial-scale=1.0">
6      <title>Edit Record</title>
7      <style>
8          body {
9              font-family: 'Segoe UI', Tahoma, Geneva, Verdana, sans-serif;
10             background: linear-gradient(to right, #ffdde1, #ee9ca7);
11             color: #333;
12             margin: 0;
13             display: flex;
14             align-items: center;
15             justify-content: center;
16             height: 100vh;
17         }
18
19         .edit-container {
20             text-align: center;
21             padding: 20px;
22             border-radius: 10px;
23             background-color: rgba(255, 255, 255, 0.9);
24             box-shadow: 0 0 20px rgba(0, 0, 0, 0.2);
25             max-width: 600px;
26             width: 100%;
27         }
28
29         h1 {
30             color: #333;
31             background: linear-gradient(to right, #ff5858, #f857a6, #8449e6, #434343);
32             padding: 15px;
33             border-radius: 10px;
```

12/5/2023

```
34     margin-bottom: 20px;
35 }
36
37 label {
38     display: block;
39     margin-top: 15px;
40     font-weight: bold;
41     color: #555;
42 }
43
44 input, select {
45     width: calc(100% - 24px);
46     padding: 12px;
47     margin-top: 8px;
48     margin-bottom: 20px;
49     box-sizing: border-box;
50     border: 1px solid #ccc;
51     border-radius: 5px;
52     font-size: 16px;
53     background-color: #f9f9f9;
54     color: #555;
55 }
56
57 input[type="submit"], .delete-link {
58     background-color: #ff5e62;
59     color: white;
60     padding: 14px 20px;
61     margin: 8px 4px;
62     border: none;
63     border-radius: 5px;
64     cursor: pointer;
65     width: 48%;
66     font-size: 16px;
```

12/5/2023

```
67     transition: background-color 0.3s;
68     display: inline-block;
69   }
70
71   .delete-link {
72     background-color: #ffb6b9;
73   }
74
75   input[type="submit"]:hover, .delete-link:hover {
76     background-color: #e04f54;
77   }
78 </style>
79 </head>
80 <body>
81   <div class="edit-container">
82     <h1>Edit Record</h1>
83
84     {% if success_message %}
85       <p style="color: #4CAF50;">{{ success_message }}</p>
86     {% endif %}
87
88     <form action="/edit_record/{{ record[0] }}" method="post">
89       <input type="hidden" name="record_id" value="{{ record[0] }}">
90
91       <!-- Display existing data for reference -->
92       <p><strong>ID:</strong> {{ record[0] }}</p>
93       <p><strong>Company:</strong> {{ record[1] }}</p>
94
95       <!-- Allow the user to edit data -->
96       <label for="new_amount">New Amount:</label>
97       <input type="text" id="new_amount" name="new_amount" value="{{ record[2] }}>
98
99       <!-- Dropdown list for the due date -->
100      <label for="due_date">Due Date:</label>
101      <select id="due_date" name="due_date" required>
102        {% for month, day in [('April', 15), ('June', 15), ('September', 15), ('January', 15)] %}
103          <option value="{{ month }} {{ day }}"{% if record[5] == month ~ ' ' ~ day %}selected{% endif %}>{{ month }} {{ day }}</option>
104        {% endfor %}
105      </select>
106
107      <!-- Add other fields as needed -->
108
109      <input type="submit" value="Save Changes">
110      <a href="#" class="delete-link" onclick="confirmDelete()">Delete</a>
111    </form>
112  </div>
113
114  <script>
115    // Function to show confirmation popup for delete
116    function confirmDelete() {
117      var result = confirm("Are you sure you want to delete this record?");
118      if (result) {
119        window.location.href = "/delete_record/{{ record[0] }}"; // Redirect to delete route
120      }
121    }
122  </script>
123 </body>
124 </html>
```

12/5/2023

## new\_user\_signup.html

```
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <meta name="viewport" content="width=device-width, initial-scale=1.0">
6      <title>New User Signup</title>
7      <style>
8          body {
9              font-family: 'Segoe UI', Tahoma, Geneva, Verdana, sans-serif;
10             background: linear-gradient(to right, #bdc3c7, #2c3e50);
11             color: #333;
12             margin: 0;
13             display: flex;
14             align-items: center;
15             justify-content: center;
16             height: 100vh;
17         }
18
19         .signup-container {
20             text-align: center;
21             padding: 20px;
22             border-radius: 10px;
23             background-color: rgba(255, 255, 255, 0.9);
24             box-shadow: 0 0 20px rgba(0, 0, 0, 0.2);
25             max-width: 400px;
26             width: 100%;
27         }
28
29         h1 {
30             color: #333;
31             background: linear-gradient(to right, #ff5858, #f857a6, #8449e6, #434343);
32             padding: 15px;
33             border-radius: 10px;
```

12/5/2023

```
34         margin-bottom: 20px;
35     }
36
37     label {
38         display: block;
39         margin-top: 15px;
40         font-weight: bold;
41         color: #555;
42     }
43
44     input {
45         width: calc(100% - 24px);
46         padding: 12px;
47         margin-top: 8px;
48         margin-bottom: 20px;
49         box-sizing: border-box;
50         border: 1px solid #ccc;
51         border-radius: 5px;
52         font-size: 16px;
53         background-color: #f9f9f9;
54         color: #555;
55     }
56
57     input[type="submit"] {
58         background-color: #4CAF50;
59         color: white;
60         padding: 14px 20px;
61         margin: 8px 4px;
62         border: none;
63         border-radius: 5px;
64         cursor: pointer;
```

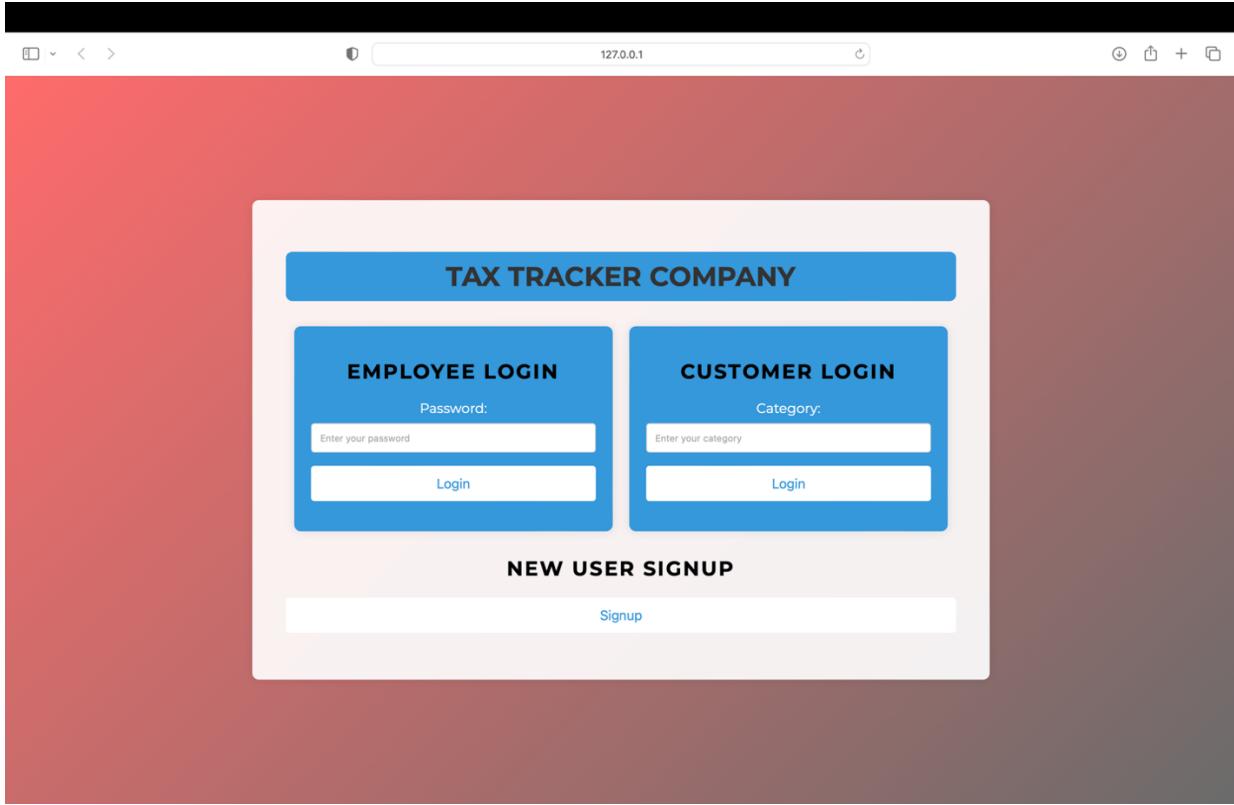
12/5/2023

```
65     width: 50%;  
66     font-size: 16px;  
67     transition: background-color 0.3s;  
68     display: inline-block;  
69   }  
70  
71   input[type="submit"]:hover {  
72     background-color: #45a049;  
73   }  
74 </style>  
75 </head>  
76 <body>  
77   <div class="signup-container">  
78     <h1>New User Signup</h1>  
79  
80     <form action="/new_user_signup" method="post">  
81       <!-- The ID is generated automatically, so no need to display it in the form -->  
82       <label for="company">Company:</label>  
83       <input type="text" id="company" name="company" required>  
84  
85       <label for="amount">Amount:</label>  
86       <input type="text" id="amount" name="amount" required>  
87  
88       <input type="submit" value="Signup">  
89     </form>  
90   </div>  
91 </body>  
92 </html>  
93
```

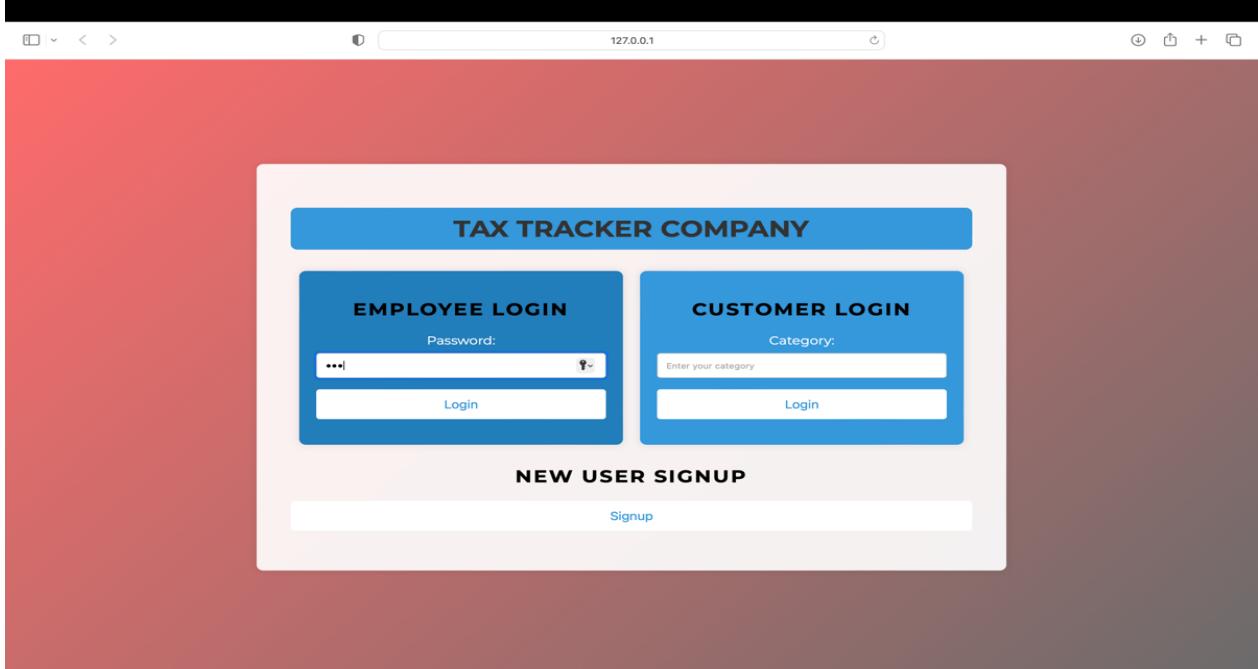
12/5/2023

## OUTPUT:

The front page looks like below:

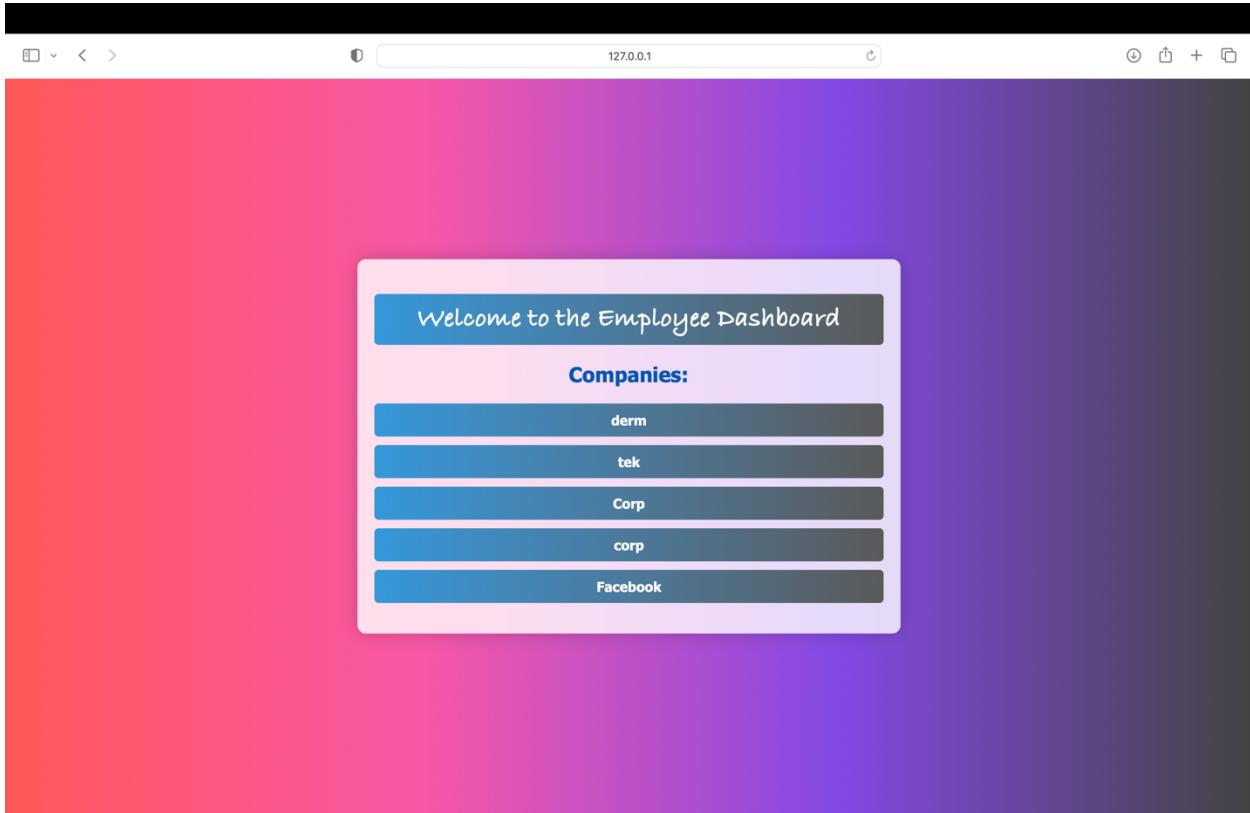


When you enter with Employee Login

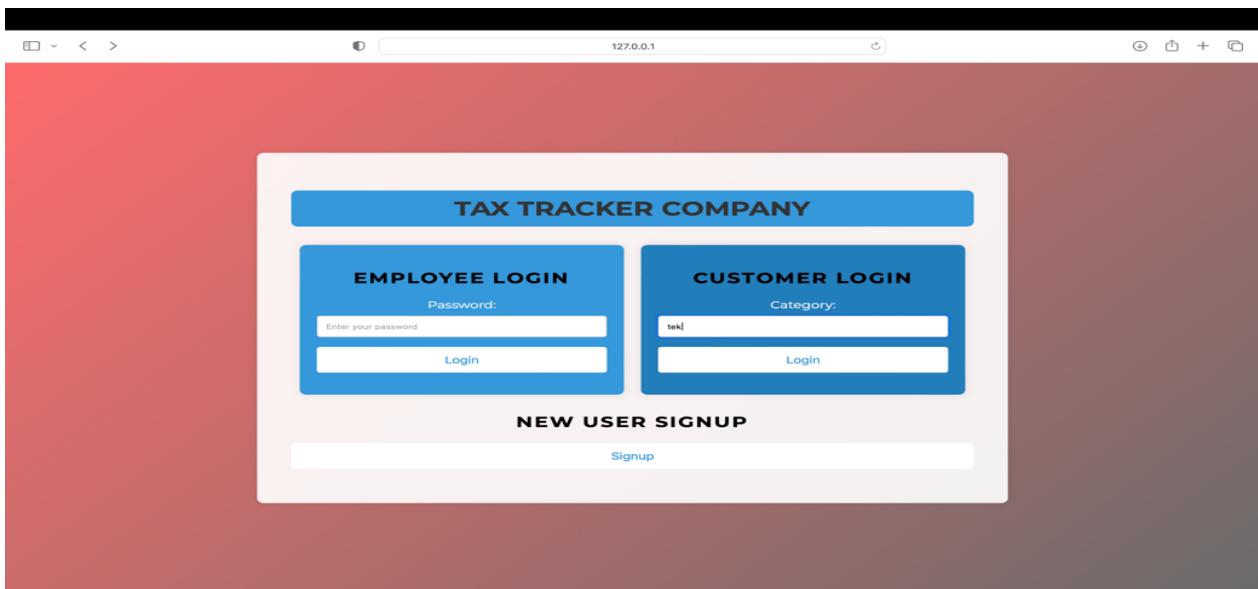


12/5/2023

You see the following page with all the registered companies



If the user wants to log in, they can log in with their company code:

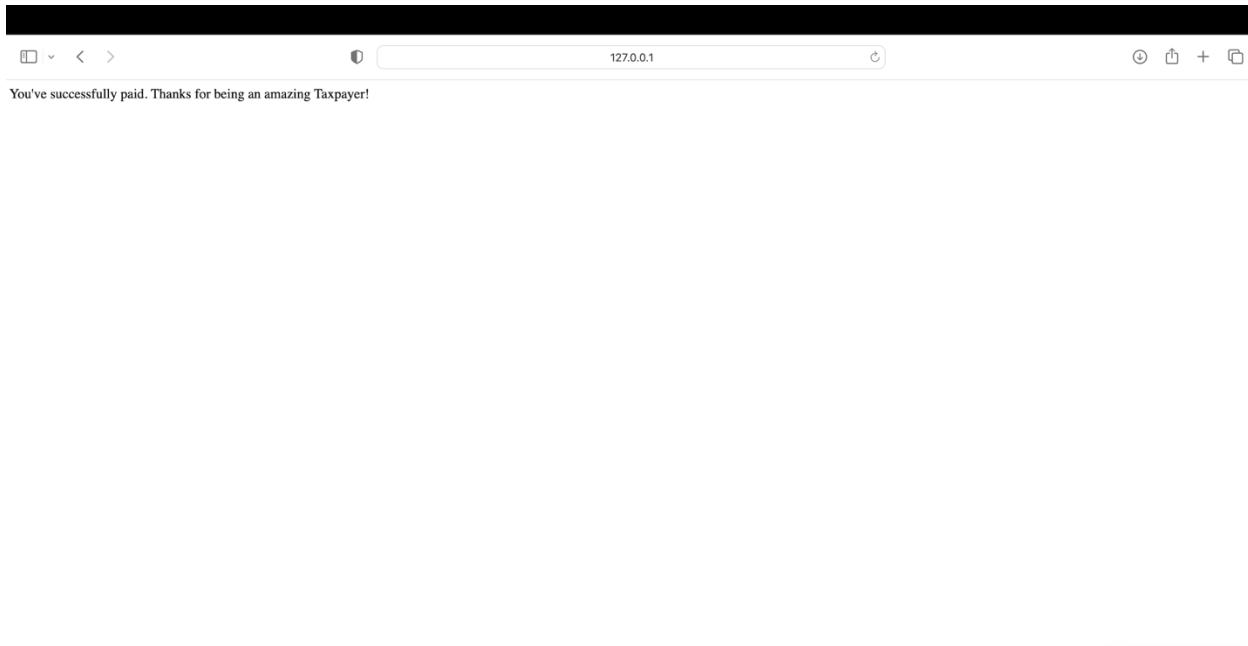


12/5/2023

And we get the following information.

127.0.0.1		
ID: 4 Company: tek Amount: 15200.0 Status: paid Date Paid: 2023-07-12 Due Date: 2024-01-15 00:00:00 <a href="#">Edit</a>	ID: 6 Company: tek Amount: 14440.0 Status: paid Date Paid: 2023-09-21 Due Date: 2024-01-15 <a href="#">Edit</a>	ID: 7 Company: tek Amount: 15200.0 Status: paid Date Paid: 2023-10-18 Due Date: 2024-01-15 <a href="#">Edit</a>
ID: 8 Company: tek Amount: 23520.0 Status: paid Due Date: 2024-01-15 <a href="#">Edit</a>	ID: 9 Company: tek Amount: 16800.0 Status: unpaid Due Date: 2024-01-15 <a href="#">Pay Here</a> <a href="#">Edit</a>	ID: 10 Company: tek Amount: 16800.0 Status: paid Due Date: 2024-01-15 <a href="#">Edit</a>
Company Information		ID: 11 Company: tek Amount: 16800.0 Status: unpaid Due Date: 2024-01-15 <a href="#">Pay Here</a>

When you click on Pay Here. You get the following message



12/5/2023

The pay here option for the following ID goes away after clicking on it

Company Information

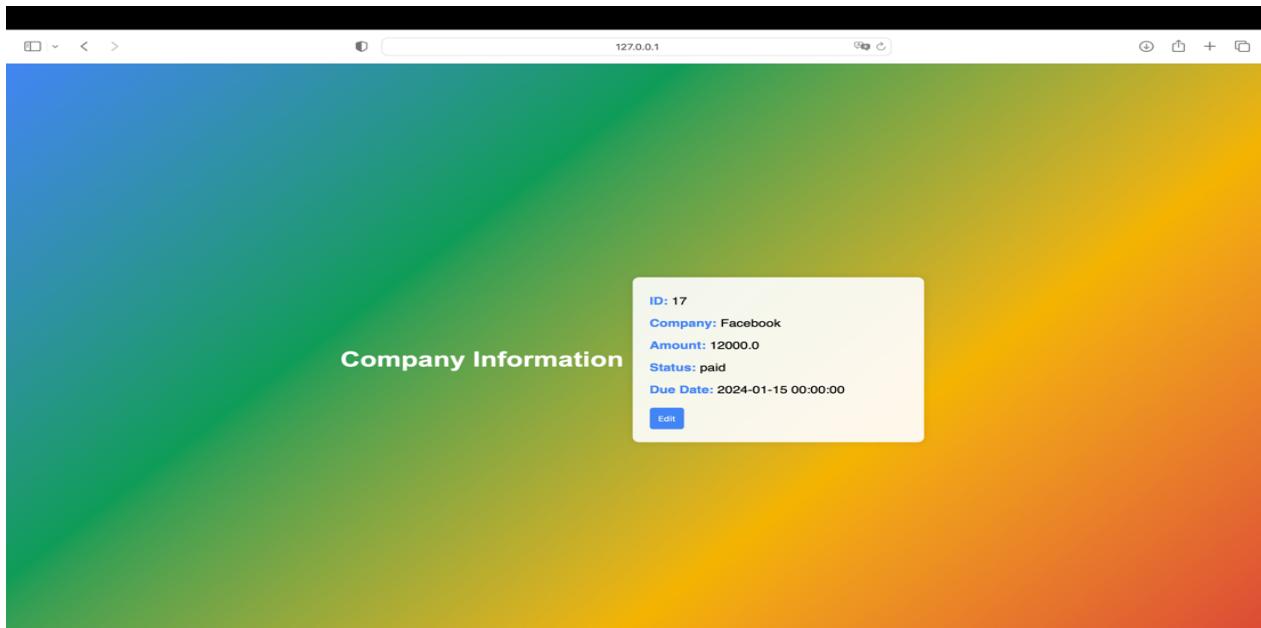
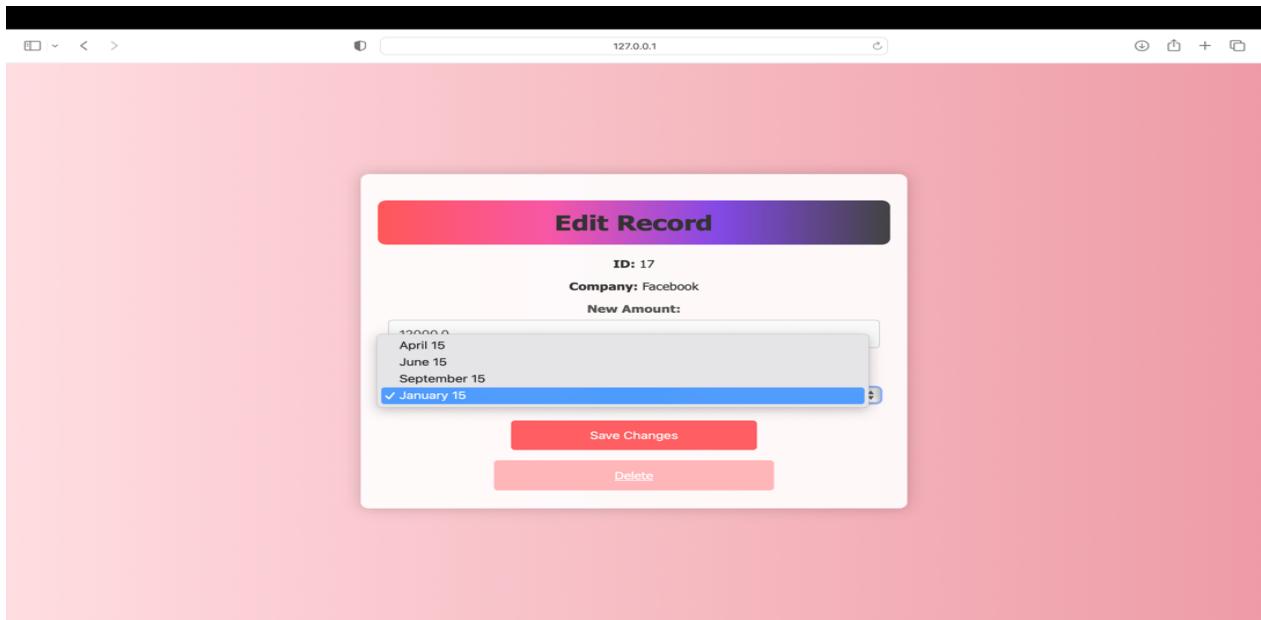
ID: 4 Company: tek Amount: 15200.0 Status: paid Date Paid: 2023-07-12 Due Date: 2024-01-15 00:00:00 <a href="#">Edit</a>	ID: 6 Company: tek Amount: 14440.0 Status: paid Date Paid: 2023-09-21 Due Date: 2024-01-15 <a href="#">Edit</a>	ID: 7 Company: tek Amount: 15200.0 Status: paid Date Paid: 2023-10-18 Due Date: 2024-01-15 <a href="#">Edit</a>
ID: 8 Company: tek Amount: 23520.0 Status: paid Due Date: 2024-01-15 <a href="#">Edit</a>	ID: 9 Company: tek Amount: 16800.0 Status: paid Due Date: 2024-01-15 <a href="#">Edit</a>	ID: 10 Company: tek Amount: 16800.0 Status: paid Due Date: 2024-01-15 <a href="#">Edit</a>
ID: 11 Company: tek Amount: 16800.0 Status: unpaid Due Date: 2024-01-15 <a href="#">Pay Here</a> <a href="#">Edit</a>		

If we click on Edit and make changes to the Due Date, it reflects like below:

Company Information

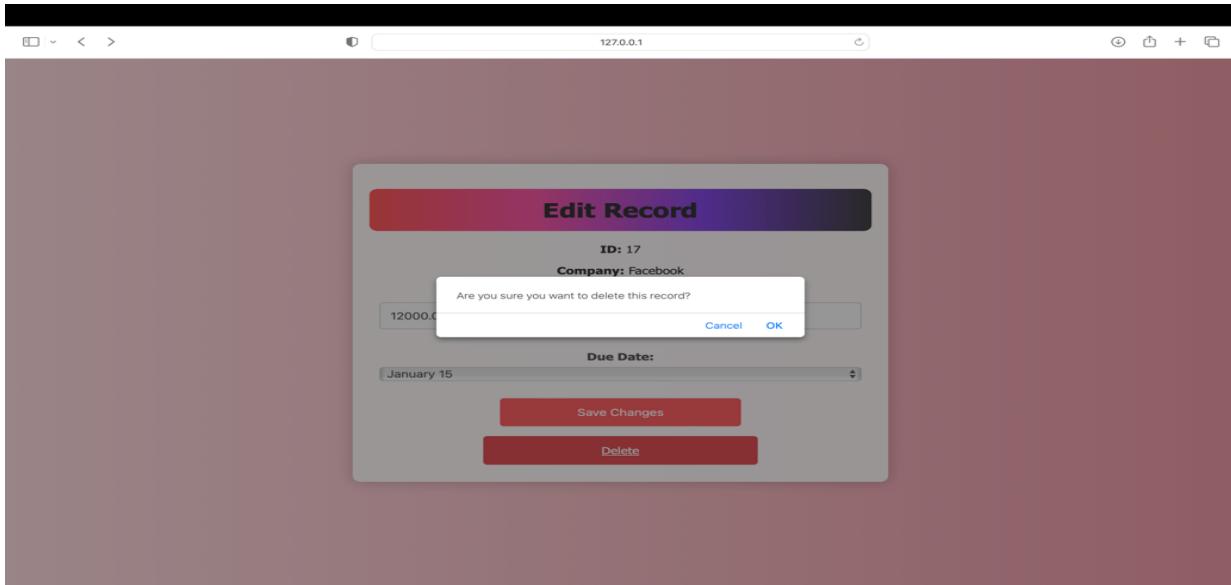
ID: 17 Company: Facebook Amount: 12000.0 Status: paid Due Date: 2023-01-15 <a href="#">Edit</a>
--

12/5/2023

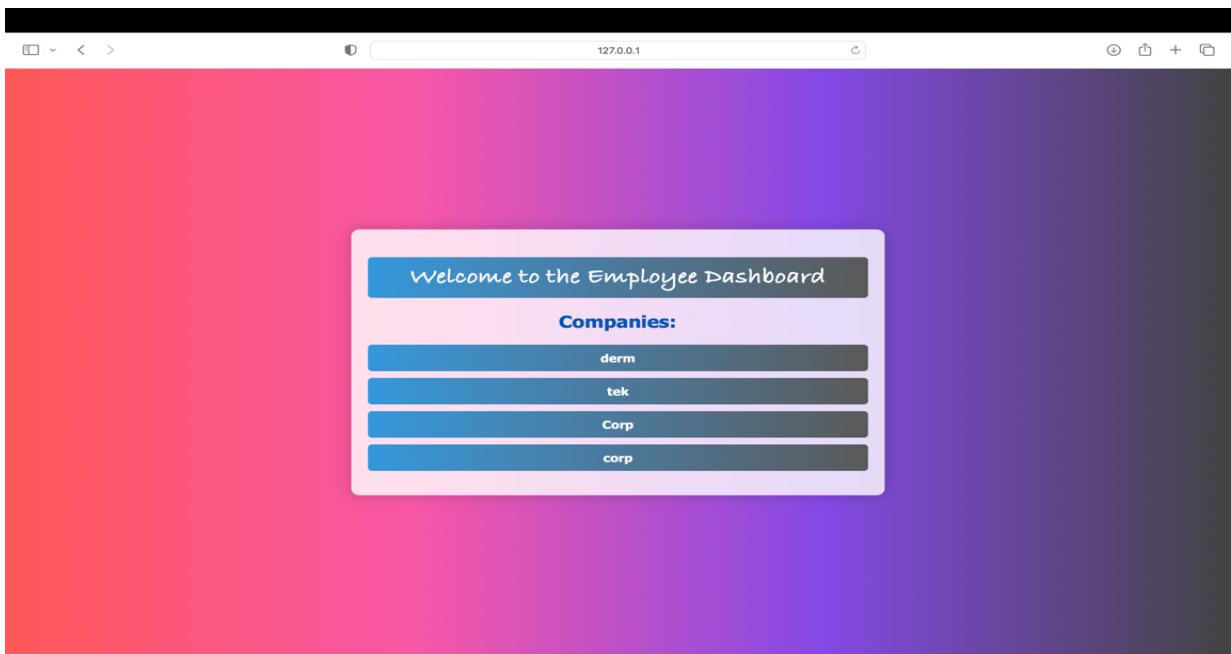


12/5/2023

If they want to delete the record, we can click on it:

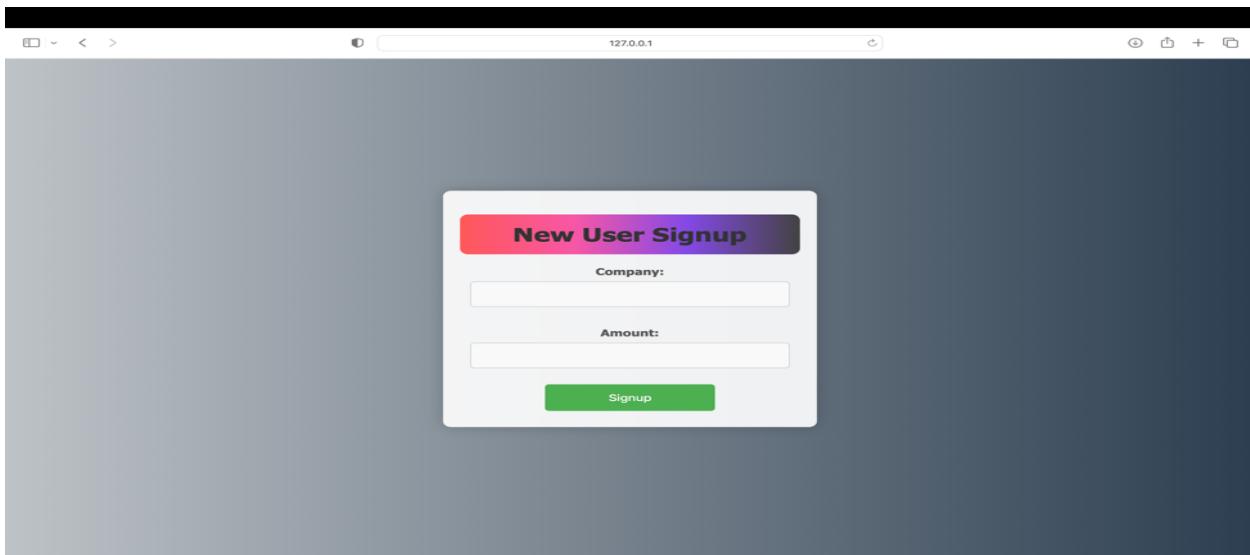
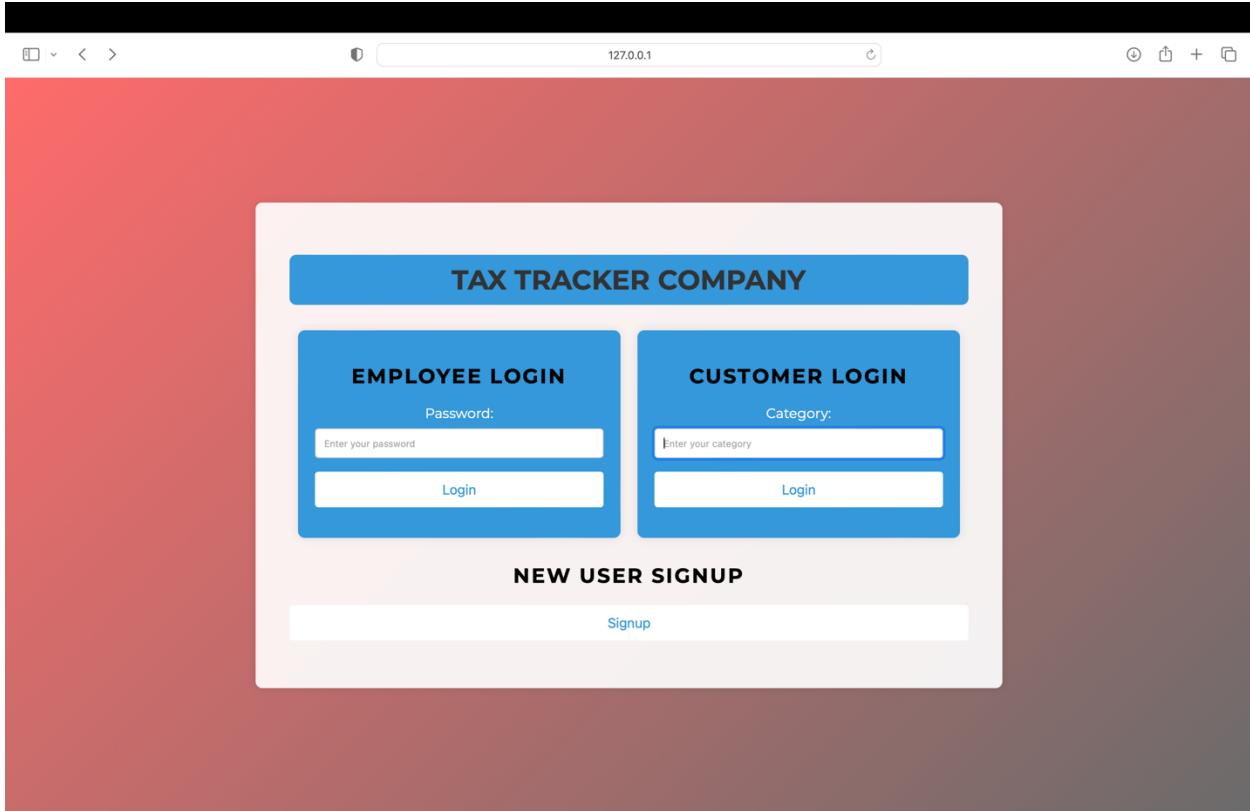


In the below you can see the Facebook is deleted from the record:

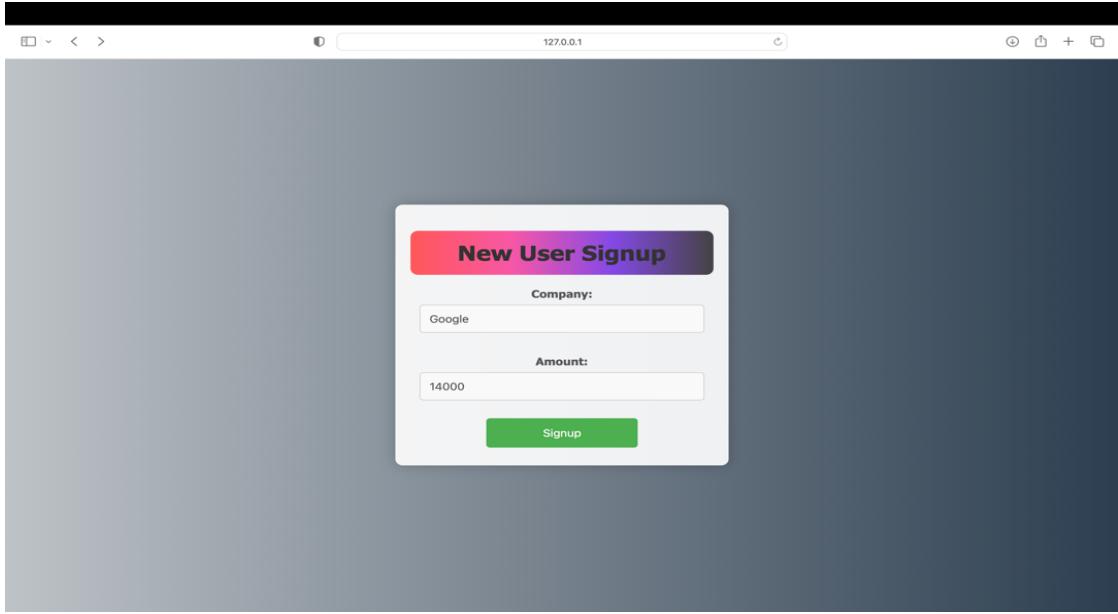


12/5/2023

For a new user signup, click on user signup—you'll directed like below as you can see in a screenshot:



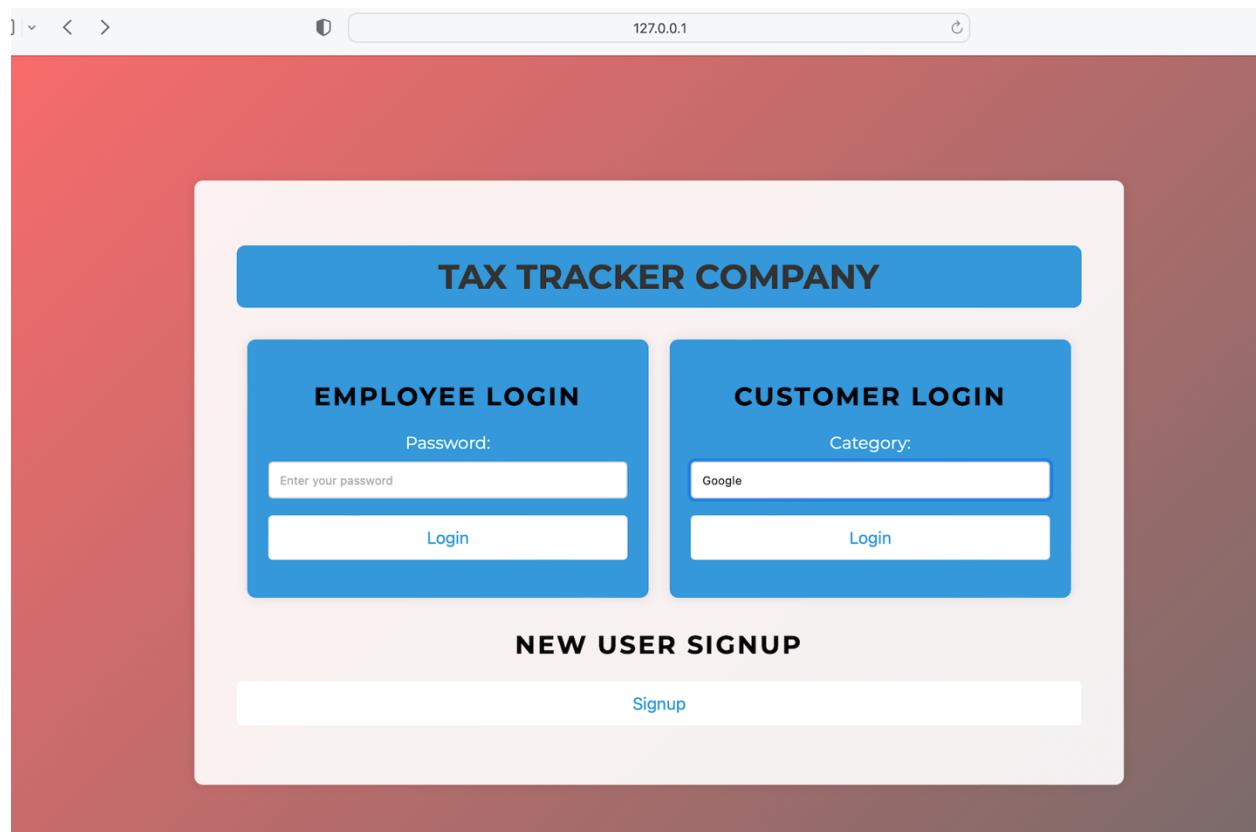
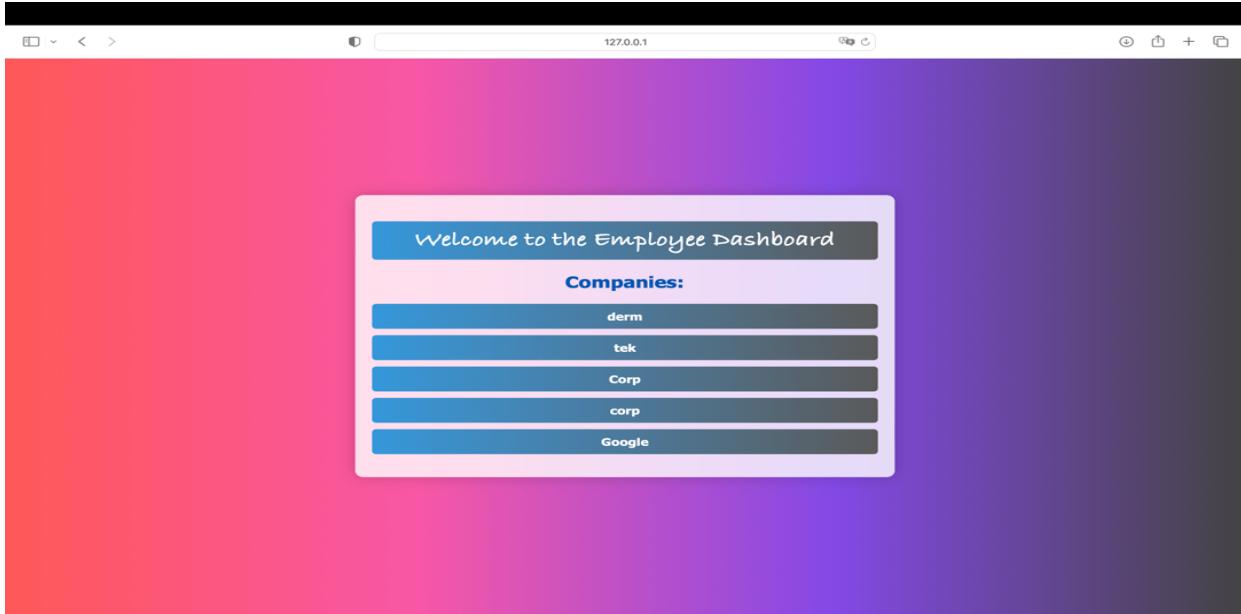
12/5/2023



A new record will be updated with a new ID



12/5/2023



12/5/2023

