

Welcome

Name:

Email:

Password:

[Login](#) [Register](#)

Users

- **Name:** Alice, **Email:** alice@example.com
- **Name:** ree, **Email:** ree@example.com
- **Name:** ad, **Email:** ad@example.com
- **Name:** Andy, **Email:** andy@example.com

Welcome

Name:

Email:

Password: 

[Login](#) [Register](#)

Users

- **Name:** Alice, **Email:** alice@example.com
- **Name:** ree, **Email:** ree@example.com
- **Name:** ad, **Email:** ad@example.com
- **Name:** Andy, **Email:** andy@example.com

Welcome

Name:

Email:

Password: 

[Login](#) [Register](#)

Users

- **Name:** Alice, **Email:** alice@example.com
- **Name:** ree, **Email:** ree@example.com
- **Name:** ad, **Email:** ad@example.com
- **Name:** Andy, **Email:** andy@example.com
- **Name:** insertsample, **Email:** insertsample@example.com

Welcome

Name:

Email:

Password:

[Login](#) [Register](#)

Successfully registered! 

Users

- **Name:** Alice, **Email:** alice@example.com
- **Name:** ree, **Email:** ree@example.com
- **Name:** ad, **Email:** ad@example.com
- **Name:** Andy, **Email:** andy@example.com
- **Name:** insertsample, **Email:** insertsample@example.com

Welcome



[Register/Login](#)

Users

- **Name:** Alice, **Email:** alice@example.com
- **Name:** ree, **Email:** ree@example.com
- **Name:** ad, **Email:** ad@example.com
- **Name:** Andy, **Email:** andy@example.com
- **Name:** insertsample, **Email:** insertsample@example.com
- **Name:** insertsample2, **Email:** insertsample2@example.com

```
94 @app.route("/register", methods=["POST"])
95 def register_user():
96     try:
97         name = request.json["name"]
98         email = request.json["email"]
99         password = request.json["password"]
100        uID = str(uuid.uuid4())
101
102        cursor = connection.cursor()
103
104        hashed_password = bcrypt.generate_password_hash(password).decode('utf-8')
105
106        try:
107            cursor.execute("INSERT INTO Users (uID, uName, Email, Passw) VALUES (%s, %s, %s, %s)", (uID, name, email, hashed_password))
108
109            connection.commit()
110            return jsonify({
111                "uName": name,
112                "Email": email,
113            }), 201
114        except psycopg2.Error as e:
115            print("Error while registering user:", e)
116            return jsonify({"error": "Failed to register user"}), 500
117
118        finally:
119            cursor.close()
120
121    except:
122        return jsonify({
123            "error": "Something is wrong. Please try again."
124        })
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
```

INSERT ^ Line 94-123 in server/main.py

Line 107 for the query

```

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

    user_id = session.get("uID")
    if request.method == 'GET':
        try:
            cursor = connection.cursor()
            cursor.execute("SELECT * FROM HaveFriend WHERE UserID = %s", (user_id,))
            friends_data = cursor.fetchall()
            cursor.close()

            friends = []
            for friend in friends_data:
                friend_dict = {
                    "UserID": friend[0],
                    "FriendID": friend[1],
                    "PlaceMet": friend[2]
                }
                friends.append(friend_dict)

            return jsonify(friends)

        except (Exception, psycopg2.Error) as error:
            print("Error fetching data:", error)
            return jsonify({"error": "Error fetching data"}), 500

    if request.method == 'POST':
        try:
            print("userID", user_id)
            new_friend_data = request.json

            cursor = connection.cursor()
            cursor.execute("SELECT uID FROM Users WHERE uName = %s AND Email = %s", (new_friend_data["Name"], new_friend_data["Email"]))
            friend_id = cursor.fetchone()
            if not friend_id:
                return jsonify({"error": "Friend not found"}), 404
            print("friend-id", friend_id[0])

            cursor.execute("INSERT INTO HaveFriend (UserID, FriendID, PlaceMet) VALUES (%s, %s, %s)",
                           (user_id, friend_id[0], new_friend_data["PlaceMet"]))

            # cursor.execute("INSERT INTO HaveFriend (UserID, FriendID, PlaceMet) VALUES (%s, %s, %s)",
            #               (friend_id[0], user_id, new_friend_data["PlaceMet"]))

            connection.commit()
            cursor.close()

            return jsonify({"message": "Friend added successfully"}), 200

        except (Exception, psycopg2.Error) as error:
            print("Error adding friend:", error)
            return jsonify({"error": "Error adding friend"}), 500

```

INSERT TO FOREIGN KEYS in server/main.py ^ line 230-282

Line 268-269 for queries

Welcome

Name:

Email:

Password:

[Login](#) [Register](#)

Users

- **Name:** Alice, **Email:** alice@example.com
- **Name:** ree, **Email:** ree@example.com
- **Name:** ad, **Email:** ad@example.com
- **Name:** Andy, **Email:** andy@example.com
- **Name:** insertsample, **Email:** insertsample@example.com
- **Name:** insertsample2, **Email:** insertsample2@example.com

[Back to Menu](#)

Profile

Name: Andy [Update Name](#)

Email: andy@example.com [Update Email](#)

[Update Password](#)

[Delete Account](#)

[Back to Menu](#)

Profile

Name: Andy [Update Name](#)

Email: andy@example.com [Update Email](#)

[Update Password](#)

Update Name

New Name:

[Submit](#)

[Delete Account](#)

localhost:5173 says

Name changed!

OK

[Back to Menu](#)

Profile

Name: Andy [Update Name](#)

Email: andy@example.com [Update Email](#)

[Update Password](#)

Update Name

New Name:

Andrew

[Submit](#)

[Delete Account](#)

[Back to Menu](#)

Profile

Name: Andrew [Update Name](#)

Email: andy@example.com [Update Email](#)

[Update Password](#)

[Delete Account](#)

[Back to Menu](#)

Profile

Name: Andrew [Update Name](#)

Email: andy@example.com [Update Email](#)

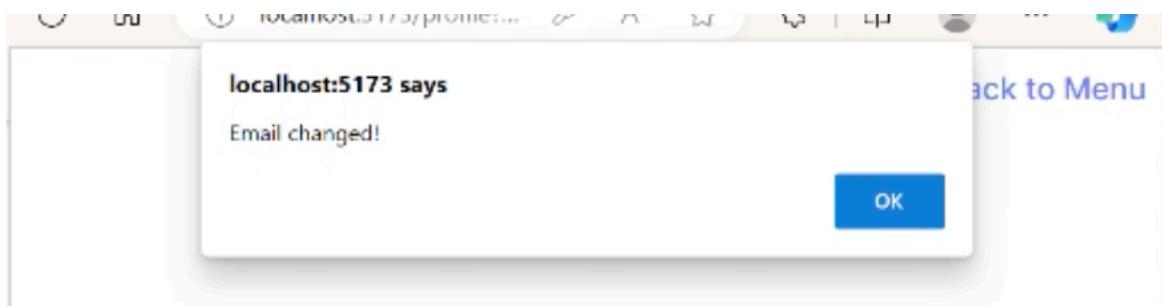
[Update Password](#)

Update Email

New Email:

[Submit](#)

[Delete Account](#)



Profile

Name: Andrew [Update Name](#)

Email: andy@example.com [Update Email](#)

[Update Password](#)

Update Email

New Email:

[Submit](#)

[Delete Account](#)

[Back to M](#)

Profile

Name: Andrew [Update Name](#)

Email: andrew@example.com [Update Email](#)

[Update Password](#)

[Delete Account](#)

[Back to Menu](#)

Profile

Name: Andrew [Update Name](#)

Email: andrew@example.com [Update Email](#)

[Update Password](#)

Update Password

Old Password:

New Password:

[Submit](#)

[Delete Account](#)

localhost:5173 says

Old Password: password123

New Password: andrew123

[Back to Menu](#)

OK

Profile

Name: Andrew [Update Name](#)

Email: andrew@example.com [Update Email](#)

[Update Password](#)



Update Password

Old Password:

New Password:

[Submit](#)

[Delete Account](#)

```

157 @app.route('/userupdate', methods = ['PUT', 'GET', 'DELETE'])
158 def userupdate():
159     try:
160         user_id = session.get("uID")
161
162         cursor = connection.cursor()
163         cursor.execute("SELECT uName, Email, Passw FROM Users WHERE uID = %s", (user_id,))
164         user = cursor.fetchone()
165
166         if not user:
167             cursor.close()
168             return jsonify({"error": "User not found"}), 404
169
170         uName, stored_email, stored_password = user
171
172         if request.method == "GET":
173             cursor.close()
174             return jsonify({
175                 "id": user_id,
176                 "name": uName,
177                 "email": stored_email
178             })
179
180         if request.method == "PUT":
181             data = request.json
182             currStatus = data.get("status")
183
184             if currStatus == "nameChange":
185                 new_name = data.get("name")
186                 if new_name:
187                     cursor.execute("UPDATE Users SET uName = %s WHERE uID = %s", (new_name, user_id))
188                     connection.commit()
189                     cursor.close()
190                     return jsonify({"message": "Name updated successfully"})
191                 else:
192                     return jsonify({"error": "New name not provided"}), 400
193
194             if currStatus == "emailChange":
195                 new_email = data.get("email")
196                 if new_email:
197                     cursor.execute("UPDATE Users SET Email = %s WHERE uID = %s", (new_email, user_id))
198                     connection.commit()
199                     cursor.close()
200                     return jsonify({"message": "Email updated successfully"})
201                 else:
202                     return jsonify({"error": "New email not provided"}), 400
203
204             if currStatus == "passwordChange":
205                 old_password = data.get("oldPassword")
206                 new_password = data.get("newPassword")
207                 if not old_password or not new_password:
208                     return jsonify({"error": "Old password and new password are required"}), 400
209                 if not bcrypt.check_password_hash(stored_password, old_password):
210                     return jsonify({"error": "Old password is incorrect"}), 401
211
212                 hashed_new_password = bcrypt.generate_password_hash(new_password).decode('utf-8')
213                 cursor.execute("UPDATE Users SET Passw = %s WHERE uID = %s", (hashed_new_password, user_id))
214                 connection.commit()
215                 cursor.close()
216                 return jsonify({"message": "Password updated successfully."})
217
218
219             if request.method == "DELETE":
220                 cursor.execute("DELETE FROM Users WHERE uID = %s", (user_id,))
221                 connection.commit()
222                 cursor.close()
223                 return jsonify({"message": "User deleted successfully"})
224
225     except:
226         return jsonify({
227             "error": "Something is wrong. Please try again."
228         })

```

UPDATE^ Line 157-228

Update queries are in Line 187, 197 and 213

Welcome

[Register/Login](#)

Users

- **Name:** Alice, **Email:** alice@example.com
- **Name:** ree, **Email:** ree@example.com
- **Name:** ad, **Email:** ad@example.com
- **Name:** Andy, **Email:** andy@example.com

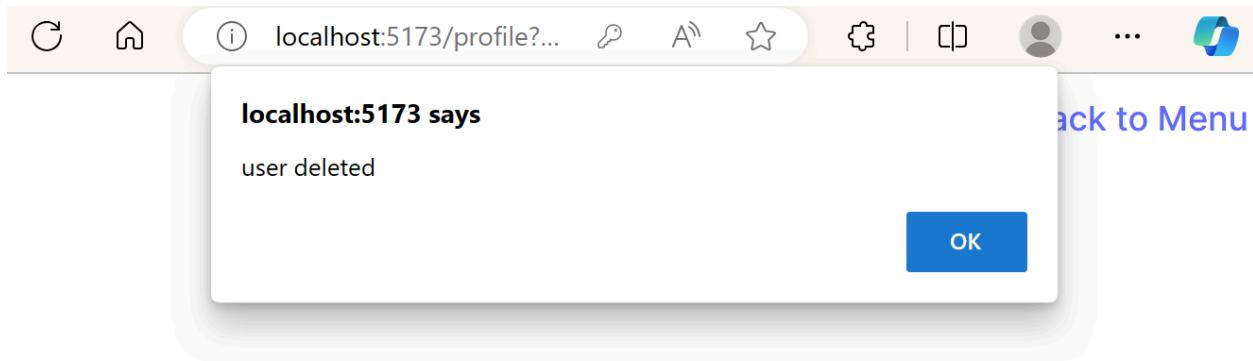
Profile

Name: ad [Update Name](#)

Email: ad@example.com [Update Email](#)

[Update Password](#)

[Delete Account](#)



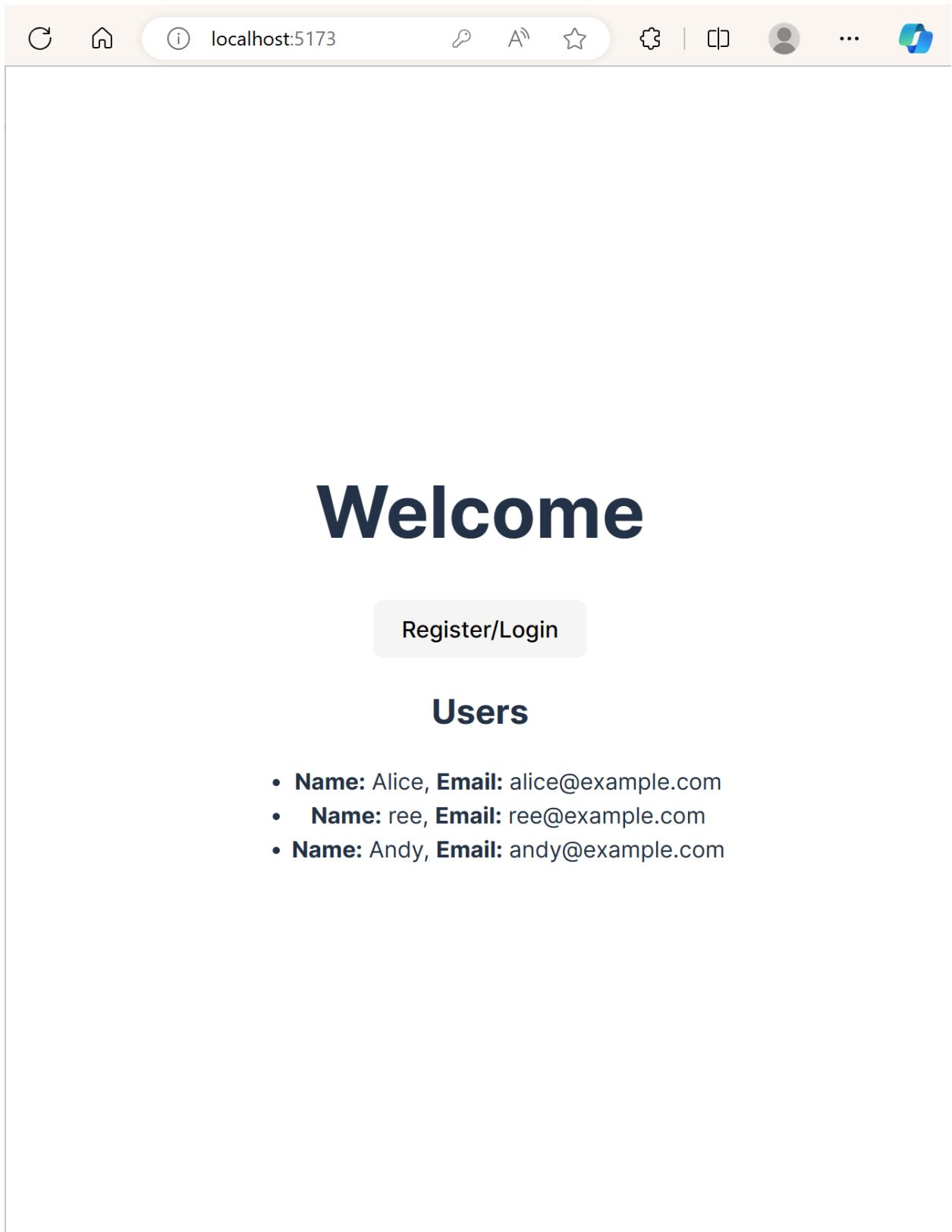
Profile

Name: ad [Update Name](#)

Email: ad@example.com [Update Email](#)

[Update Password](#)

[Delete Account](#)



```

@app.route('/userupdate', methods = ['PUT', 'GET', 'DELETE'])
def userupdate():
    try:
        user_id = session.get("uID")

        cursor = connection.cursor()
        cursor.execute("SELECT uName, Email, Passw FROM Users WHERE uID = %s", (user_id,))
        user = cursor.fetchone()

        if not user:
            cursor.close()
            return jsonify({"error": "User not found"}), 404

        uName, stored_email, stored_password = user

        if request.method == "GET":
            cursor.close()
            # print("userupdate userId", user_id)
            return jsonify({
                "id": user_id,
                "name": uName,
                "email": stored_email
            })

        if request.method == "PUT":
            data = request.json
            currStatus = data.get("status")

            if currStatus == "nameChange":
                new_name = data.get("name")
                if new_name:
                    cursor.execute("UPDATE Users SET uName = %s WHERE uID = %s", (new_name, user_id))
                    connection.commit()
                    cursor.close()
                    return jsonify({"message": "Name updated successfully"})
                else:
                    return jsonify({"error": "New name not provided"}), 400

            if currStatus == "emailChange":
                new_email = data.get("email")
                if new_email:
                    cursor.execute("UPDATE Users SET Email = %s WHERE uID = %s", (new_email, user_id))
                    connection.commit()
                    cursor.close()
                    return jsonify({"message": "Email updated successfully"})
                else:
                    return jsonify({"error": "New email not provided"}), 400

            if currStatus == "passwordChange":
                old_password = data.get("oldPassword")
                new_password = data.get("newPassword")
                if not old_password or not new_password:
                    return jsonify({"error": "Old password and new password are required"}), 400
                if not bcrypt.check_password_hash(stored_password, old_password):
                    return jsonify({"error": "Old password is incorrect"}), 401

                hashed_new_password = bcrypt.generate_password_hash(new_password).decode('utf-8')
                cursor.execute("UPDATE Users SET Passw = %s WHERE uID = %s", (hashed_new_password, user_id))
                connection.commit()
                cursor.close()
                return jsonify({"message": "Password updated successfully."})

        if request.method == "DELETE":
            cursor.execute("DELETE FROM Users WHERE uID = %s", (user_id,))
            connection.commit()
            cursor.close()
            return jsonify({"message": "User deleted successfully"})

    except:
        return jsonify({
            "error": "Something is wrong. Please try again."
        })
    
```

DELETE in server/main.py^ from line 219 to line 228. The query is on line 220.

[Home](#)

[Calendar](#)

[Diary](#)

[List](#)

[Issue Board](#)
[Emotion Board](#)

There are 5 fields to choose from in the calendar:

y.yearfield,y.Yeartheme,y.Summary,yt.Theme

Example search: y.yearfield,y.Yeartheme,y.Summary,

Example search: y.yearfield = 2022 AND y.Yeartheme = "Hello World"

havelssue: NO

number of issues by year overall

number of emotions by year overall

number of issues within the recent ? year

number of issues within ? year resolved

find all the unresolved or resolved issues that exists every year.

Home
Calendar Diary
List

Issue Board Emotion Board

There are 5 fields to choose from in the calendar:

y.yearfield,y.Yeartheme,y.Summary,yt.Theme

Example search: y.yearfield,y.Yeartheme,y.Summary,

Projection

ProjectionTable

Example search: y.yearfield = 2022 AND y.Yeartheme = "Hello World"

y.yearfield = 2022

Search

havelIssue: NO

number of issues by year overall

number of emotions by year overall

number of issues within the recent ? year

Enter year

number of issues within ? year resolved

Enter year

find all the unresolved or resolved issues that exists every year.

Issue Resolved: NO

Use YearTheme

Home
Calendar Diary Goal
List

Issue Board Emotion Board

There are 5 fields to choose from in the calendar:

y.yearfield,y.Yeartheme,y.Summary,yt.Theme

Example search: y.yearfield,y.Yeartheme,y.Summary,

Projection ProjectionTable

Example search: y.yearfield = 2022 AND y.Yeartheme = "Hello World"

Search

havelsue: NO

number of issues by year overall

number of emotions by year overall

number of issues within the recent ? year

number of issues within ? year resolved

find all the unresolved or resolved issues that exists every year. Issue Resolved: NO

Use YearTheme

```
[["andy@example.com", 2022, "Second Year with Honours", "Academics", "Cpsc 210 my beloved",  
"691e766a-714c-4512-a650-ecf17b40227d"], ["andy@example.com", 2022, "Third Year with  
Honours", "Academics", "I do not like finance", "691e766a-714c-4512-a650-ecf17b40227d"],  
["andy@example.com", 2022, "Third Year with Honours", "Academics", "I do not like hacks",  
"691e766a-714c-4512-a650-ecf17b40227d"], ["andy@example.com", 2022, "Third Year with  
Honours", "Academics", "I love studying", "691e766a-714c-4512-a650-ecf17b40227d"],  
["andy@example.com", 2022, "Third Year with Honours", "Academics", "I do not like finance",  
"691e766a-714c-4512-a650-ecf17b40227d"], ["andy@example.com", 2022, "Third Year with  
Honours", "Academics", "I do not like hacks", "691e766a-714c-4512-a650-ecf17b40227d"],  
["andy@example.com", 2022, "Third Year with Honours", "Academics", "I love studying", "691e766a-  
714c-4512-a650-ecf17b40227d"], ["andy@example.com", 2022, "Fourth Year with Honours",  
"Health and Fitness", "I love movies", "691e766a-714c-4512-a650-ecf17b40227d"],  
["andy@example.com", 2022, "Third Year with Honours", "Academics", "I do not like finance",  
"691e766a-714c-4512-a650-ecf17b40227d"], ["andy@example.com", 2022, "Third Year with  
Honours", "Academics", "I do not like hacks", "691e766a-714c-4512-a650-ecf17b40227d"],  
["andy@example.com", 2022, "Third Year with Honours", "Academics", "I love studying", "691e766a-  
714c-4512-a650-ecf17b40227d"]]
```

[Home](#)
[Calendar](#) [List](#) [Diary](#) [Goal](#)

[Issue Board](#) [Emotion Board](#)

There are 5 fields to choose from in the calendar:

y.yearfield,y.Yeartheme,y.Summary,yt.Theme

Example search: y.yearfield,y.Yeartheme,y.Summary,

Example search: y.yearfield = 2022 AND y.Yeartheme = "Hello World"

havellIssue: NO

number of issues by year overall

number of emotions by year overall

number of issues within the recent ? year

number of issues within ? year resolved

find all the unresolved or resolved issues that exists every year. Issue Resolved: NO

Use YearTheme

```
[[{"andy@example.com", 2022, "Second Year with Honours", "Academics", "Cpsc 210 my beloved", "691e766a-714c-4512-a650-ecf17b40227d"}, {"andy@example.com", 2022, "Third Year with Honours", "Academics", "I do not like finance", "691e766a-714c-4512-a650-ecf17b40227d"}, {"andy@example.com", 2022, "Third Year with Honours", "Academics", "I do not like hacks", "691e766a-714c-4512-a650-ecf17b40227d"}, {"andy@example.com", 2022, "Third Year with Honours", "Academics", "I do not like hacks", "691e766a-714c-4512-a650-ecf17b40227d"}, {"andy@example.com", 2022, "Third Year with Honours", "Academics", "I love studying", "691e766a-714c-4512-a650-ecf17b40227d"}, {"andy@example.com", 2022, "Third Year with Honours", "Academics", "I do not like finance", "691e766a-714c-4512-a650-ecf17b40227d"}, {"andy@example.com", 2022, "Third Year with Honours", "Academics", "I do not like hacks", "691e766a-714c-4512-a650-ecf17b40227d"}, {"andy@example.com", 2022, "Third Year with Honours", "Academics", "I love studying", "691e766a-714c-4512-a650-ecf17b40227d"}, {"andy@example.com", 2022, "Fourth Year with Honours", "Health and Fitness", "I love movies", "691e766a-714c-4512-a650-ecf17b40227d"}, {"andy@example.com", 2022, "Third Year with Honours", "Academics", "I do not like finance", "691e766a-714c-4512-a650-ecf17b40227d"}, {"andy@example.com", 2022, "Third Year with Honours", "Academics", "I do not like hacks", "691e766a-714c-4512-a650-ecf17b40227d"}, {"andy@example.com", 2022, "Third Year with Honours", "Academics", "I love studying", "691e766a-714c-4512-a650-ecf17b40227d"}]]
```

[Home](#)
[Calendar](#) [Diary](#) [Goal](#)
[List](#)

[Issue Board](#) [Emotion Board](#)

There are 5 fields to choose from in the calendar:

y.yearfield,y.Yeartheme,y.Summary,yt.Theme

Example search: y.yearfield,y.Yeartheme,y.Summary,

Example search: y.yearfield = 2022 AND y.Yeartheme = "Hello World"

havetIssue: NO

number of issues by year overall

number of emotions by year overall

number of issues within the recent ? year

number of issues within ? year resolved

find all the unresolved or resolved issues that exists every year. Issue Resolved: NO

```
[[{"andy@example.com", 2022, "Second Year with Honours", "Academics", "Cpsc 210 my beloved", "691e766a-714c-4512-a650-ecf17b40227d"}, {"andy@example.com", 2022, "Third Year with Honours", "Academics", "I do not like finance", "691e766a-714c-4512-a650-ecf17b40227d"}, {"andy@example.com", 2022, "Third Year with Honours", "Academics", "I do not like hacks", "691e766a-714c-4512-a650-ecf17b40227d"}, {"andy@example.com", 2022, "Third Year with Honours", "Academics", "I love studying", "691e766a-714c-4512-a650-ecf17b40227d"}, {"andy@example.com", 2022, "Third Year with Honours", "Academics", "I do not like finance", "691e766a-714c-4512-a650-ecf17b40227d"}, {"andy@example.com", 2022, "Third Year with Honours", "Academics", "I do not like hacks", "691e766a-714c-4512-a650-ecf17b40227d"}, {"andy@example.com", 2022, "Third Year with Honours", "Academics", "I love studying", "691e766a-714c-4512-a650-ecf17b40227d"}, {"andy@example.com", 2022, "Fourth Year with Honours", "Health and Fitness", "I love movies", "691e766a-714c-4512-a650-ecf17b40227d"}, {"andy@example.com", 2022, "Third Year with Honours", "Academics", "I do not like finance", "691e766a-714c-4512-a650-ecf17b40227d"}, {"andy@example.com", 2022, "Third Year with Honours", "Academics", "I do not like hacks", "691e766a-714c-4512-a650-ecf17b40227d"}, {"andy@example.com", 2022, "Third Year with Honours", "Academics", "I love studying", "691e766a-714c-4512-a650-ecf17b40227d}]]
```

Home
Calendar Diary Goal
List

Issue Board **Emotion Board**

There are 5 fields to choose from in the calendar:

y.yearfield,y.Yeartheme,y.Summary,yt.Theme

Example search: y.yearfield,y.Yeartheme,y.Summary,

Example search: y.yearfield = 2022 AND y.Yeartheme = "Hello World"

'Second Year with Honours' Search

havelissue: NO

number of issues by year overall

number of emotions by year overall

number of issues within the recent ? year

number of issues within ? year resolved

find all the unresolved or resolved issues that exists every year. Issue Resolved: NO

Use YearTheme

```
[[{"andy@example.com", 2022, "Second Year with Honours", "Academics", "Cpsc 210 my beloved", "691e766a-714c-4512-a650-ecf17b40227d"}, {"andy@example.com", 2022, "Third Year with Honours", "Academics", "I do not like finance", "691e766a-714c-4512-a650-ecf17b40227d"}, {"andy@example.com", 2022, "Third Year with Honours", "Academics", "I do not like hacks", "691e766a-714c-4512-a650-ecf17b40227d"}, {"andy@example.com", 2022, "Third Year with Honours", "Academics", "I love studying", "691e766a-714c-4512-a650-ecf17b40227d"}, {"andy@example.com", 2022, "Third Year with Honours", "Academics", "I do not like finance", "691e766a-714c-4512-a650-ecf17b40227d"}, {"andy@example.com", 2022, "Third Year with Honours", "Academics", "I do not like hacks", "691e766a-714c-4512-a650-ecf17b40227d"}, {"andy@example.com", 2022, "Third Year with Honours", "Academics", "I love studying", "691e766a-714c-4512-a650-ecf17b40227d"}, {"andy@example.com", 2022, "Fourth Year with Honours", "Health and Fitness", "I love movies", "691e766a-714c-4512-a650-ecf17b40227d"}, {"andy@example.com", 2022, "Third Year with Honours", "Academics", "I do not like finance", "691e766a-714c-4512-a650-ecf17b40227d"}, {"andy@example.com", 2022, "Third Year with Honours", "Academics", "I do not like hacks", "691e766a-714c-4512-a650-ecf17b40227d"}, {"andy@example.com", 2022, "Third Year with Honours", "Academics", "I love studying", "691e766a-714c-4512-a650-ecf17b40227d"}]]
```

Home
Calendar Diary Goal
List

Issue Board Emotion Board

There are 5 fields to choose from in the calendar:

y.yearfield,y.Yeartheme,y.Summary,yt.Theme

Example search: y.yearfield,y.Yeartheme,y.Summary,

Example search: y.yearfield = 2022 AND y.Yeartheme = "Hello World"

havelIssue: NO

number of issues by year overall

number of emotions by year overall

number of issues within the recent ? year

number of issues within ? year resolved

find all the unresolved or resolved issues that exists every year. Issue Resolved: NO

`[[{"andy@example.com", 2022, "Second Year with Honours", "Academics", "Cpsc 210 my beloved",
"691e766a-714c-4512-a650-ecf17b40227d}]]`

```

309 elif ("selection" in options):
310     projection = "SELECT u.Email as Email, y.yearfield as Year, y.Yeartheme as Yeartheme, yt.Theme as Theme, y.Summary as Summary, " + 'y.UserID as UserID'
311     selection = "(" + request.json["selection"] + ")"
312     verification = ".Email = " + request.json["email"] + " "
313     11 = " .UserID = u.uID AND u.uID = y.UserID AND i.UserID = u.uID "
314     12 = " AND yt.theme = m.Theme "
315     13 = " AND yt.YearTheme = y.YearTheme AND yt.yearfield = y.yearfield AND yt.yearID = y.yearID"
316     14 = " AND y.yearfield = idy.yearfield AND idy.yeartheme = y.yeartheme AND idy.yearID = y.yearID"
317     15 = " AND EB.yearfield = y.yearfield AND EB.yeartheme = y.yeartheme AND EB.UserID = u.uID"
318     16 = " AND ebt.Subtypes = EB.Subtypes AND ebt EmotionID = EB.EmotionID"
319     17 = " AND idy.issueID = i.issueID "
320     connectingtable = 11 + 12 + 13 + 14 + 15 + 16 + 17
321     query = projection + ' FROM YearTheme yt, Menu m, Users u, Years y, IssueDateYear idy, EB, Issue i, EBType ebt' + ' WHERE ' + connectingtable + ' AND ' + verification + "AND " + selection + ";"
322     cursor = connection.cursor()
323     cursor.execute(query)
324     data = cursor.fetchall()
325     result = json.dumps(data)

```

SELECTION ^ line 310 to 325 in server/main.py. The query is from line 310 to line 320.

[Home](#)

[Calendar](#)
[List](#)
[Diary](#)
[Goal](#)

[Issue Board](#)
[Emotion Board](#)

There are 5 fields to choose from in the calendar:

y.yearfield,y.Yeartheme,y.Summary,yt.Theme

Example search: y.yearfield,y.Yeartheme,y.Summary,

i.issueName	Issue i
-------------	---------

Example search: y.yearfield = 2022 AND y.Yeartheme = "Hello World"

Selection	Search
-----------	--------

havellssue: NO

number of issues by year overall

number of emotions by year overall

number of issues within the recent ? year

number of issues within ? year resolved

find all the unresolved or resolved issues that exists every year. Issue Resolved: NO

Use YearTheme

Home
Calendar Diary Goal
List

Issue Board **Emotion Board**

There are 5 fields to choose from in the calendar:

y.yearfield,y.Yeartheme,y.Summary,yt.Theme

Example search: y.yearfield,y.Yeartheme,y.Summary,

Example search: y.yearfield = 2022 AND y.Yeartheme = "Hello World"

Search

havelIssue: NO

number of issues by year overall

number of emotions by year overall

number of issues within the recent ? year

number of issues within ? year resolved

find all the unresolved or resolved issues that exists every year. Issue Resolved: NO

Use YearTheme

[["Project Deadline"], ["Research Challenges"], ["Career Transition"], ["Academic Concession"], ["Academic Concession"], ["Academic Concession"], ["Academic Concession"], ["Academic Concession"], ["Broke"], ["Academic Concession"], ["Academic Concession"], ["Academic Concession"], ["Academic Concession"], ["Challenge"], ["Teamwork"], ["Challenge"], ["Teamwork"], ["Challenge"], ["Challenge"], ["Challenge"], ["Teamwork"], ["Teamwork"], ["Hackathon"], ["Hackathon"], ["Hackathon"], ["Hackathon"], ["Hospital"], ["Movie"], ["Academic Concession"], ["Academic Concession"], ["Academic Concession"]]

[Home](#)
[Calendar](#) [Diary](#) [Goal](#)
[List](#)

[Issue Board](#) [Emotion Board](#)

There are 5 fields to choose from in the calendar:

y.yearfield,y.Yeartheme,y.Summary,yt.Theme

Example search: y.yearfield,y.Yeartheme,y.Summary,

Example search: y.yearfield = 2022 AND y.Yeartheme = "Hello World"

havelsue: NO

number of issues by year overall

number of emotions by year overall

number of issues within the recent ? year

number of issues within ? year resolved

find all the unresolved or resolved issues that exists every year. Issue Resolved: NO

Use YearTheme

```
[[{"Project Deadline": false}, {"Research Challenges": false}, {"Career Transition": false}, {"Academic Concession": false}, {"Academic Concession": false}, {"Academic Concession": false}, {"Academic Concession": false}, {"Broke": false}, {"Academic Concession": false}, {"Academic Concession": false}, {"Academic Concession": false}, {"Academic Concession": false}, {"Challenge": false}, {"Teamwork": true}, {"Challenge": false}, {"Teamwork": true}, {"Challenge": false}, {"Challenge": false}, {"Teamwork": true}, {"Teamwork": true}, {"Hackathon": true}, {"Hackathon": true}, {"Hackathon": true}, {"Hospital": false}, {"Movie": true}, {"Academic Concession": false}, {"Academic Concession": false}, {"Academic Concession": false}]]
```

Home
Calendar Diary Goal
List

Issue Board Emotion Board

There are 5 fields to choose from in the calendar:

y.yearfield,y.Yeartheme,y.Summary,yt.Theme

Example search: y.yearfield,y.Yeartheme,y.Summary,

i.issueName,i.details Issue i

Example search: y.yearfield = 2022 AND y.Yeartheme = "Hello World"

Selection Search

havelsue: NO

number of issues by year overall

number of emotions by year overall

number of issues within the recent ? year Enter year

number of issues within ? year resolved Enter year

find all the unresolved or resolved issues that exists every year. Issue Resolved: NO

Use YearTheme

[["Project Deadline", "Facing tight deadlines at work"], ["Research Challenges", "Encountering difficulties in data analysis"], ["Career Transition", "Uncertainty about future career path"], ["Academic Concession", "Not feeling well today"], ["Broke", "Wdym tickets are \$1000 now"], ["Academic Concession", "Not feeling well today"], ["Challenge", "Cpsc 213 assignment is too hard"], ["Teamwork", "had a fight with my teammate today"], ["Challenge", "Cpsc 304 assignment is too hard"], ["Teamwork", "had a fight with my teammate today"], ["Challenge", "Cpsc 110 assignment is too hard"], ["Challenge", "Cpsc 404 assignment is too hard"], ["Teamwork", "had a fight with my colleagues today"], ["Teamwork", "had a fight with my dog today"], ["Hackathon", "had a demo today"], ["Hackathon", "had a demo today"], ["Hackathon", "had a demo today"], ["Hospital", "had a coma today"], ["Movie", "had a movie today"], ["Academic Concession", "had a concession today"], ["Academic Concession", "had a concession today"], ["Academic Concession", "had a concession today"]]

```

326     result = json.dumps(data)
327     elif ("projection" in options):
328         projection = "SELECT " + request.json["projection"] + " "
329         verification = " WHERE u.Email = " + request.json["email"] + "" + " AND "
330         if ("projectiontable" in request.json):
331             i1 = ".m.UserID = u.UserID AND u.UserID = y.UserID AND i.UserID = u.UserID "
332             i2 = ". AND yt.Theme = m.Theme"
333             i3 = ". AND yt.YearTheme = y.YearTheme AND yt.yearfield = y.yearfield AND yt.yearID = y.yearID"
334             i4 = ". AND y.yearfield = idy.yearField AND idy.yeartheme = y.yeartheme AND idy.yearID = y.yearID"
335             i5 = ". AND EB.yearfield = y.yearfield AND EB.yeartheme = y.yeartheme AND EB.UserID = u.UserID"
336             i6 = ". AND ebt.Subtypes = EB.Subtypes AND ebt.EmotionID = EB.EmotionID"
337             i7 = ". AND idy.issueID = i.issueID "
338
339         query = projection + " FROM " + request.json["projectiontable"]
340         # query = projection + " FROM YearTheme yt, Menu m, Users u, Years y, IssueDateYear idy, EB, Issue i, EBType ebt" + " WHERE " + connectingtable + " AND " + verification + ";"
341         cursor = connection.cursor()
342         cursor.execute(query)
343         data = cursor.fetchall()
344         result = json.dumps(data)

```

```

31
32     def admin():
33         connection = psycopg2.connect(database = db_name, host = host, password = password, port = 5433, user = username)
34         try:
35             q1 = "WITH tablename(tab) AS (SELECT table_name AS tab FROM information_schema.tables WHERE table_schema='public') "
36             q2 = " SELECT tablename.tab AS TABLE, column_name FROM INFORMATION_SCHEMA.COLUMNS, tablename WHERE table_name = tablename.tab"
37             q3 = " GROUP BY tablename.tab, column_name"
38             query = q1 + q2 + q3
39             cursor = connection.cursor()
40             cursor.execute(query)
41             data = cursor.fetchall()
42             result = json.dumps(data)
43             cursor.close()
44             return jsonify({
45                 "result": result
46             })
47         # except:
48         #     return jsonify({
49         #         "error": "There is an error"
50         # })
51

```

PROJECTION ^ the function is a line 326 to line 343 ,query line is 326 to 338 in server/main.py (not the code commented out). This is used in addition with the rendering of all the tables names and their schemas in line 32 to line 40

[Home](#)

[Calendar](#)
[List](#)
[Diary](#)
[Goal](#)

[Issue Board](#)
[Emotion Board](#)

There are 5 fields to choose from in the calendar:

y.yearfield,y.Yeartheme,y.Summary,yt.Theme

Example search: y.yearfield,y.Yeartheme,y.Summary,

Example search: y.yearfield = 2022 AND y.Yeartheme = "Hello World"

havelsue: NO

number of issues by year overall

number of emotions by year overall

number of issues within the recent ? year

number of issues within ? year resolved

find all the unresolved or resolved issues that exists every year. **Issue Resolved: NO**

Use YearTheme

```
[[{"Project Deadline": "Facing tight deadlines at work"}, {"Research Challenges": "Encountering difficulties in data analysis"}, {"Career Transition": "Uncertainty about future career path"}, {"Academic Concession": "Not feeling well today"}, {"Broke": "Wdym tickets are $1000 now"}, {"Academic Concession": "Not feeling well today"}, {"Challenge": "Cpsc 213 assignment is too hard"}, {"Teamwork": "had a fight with my teammate today"}, {"Challenge": "Cpsc 304 assignment is too hard"}, {"Teamwork": "had a fight with my teammate today"}, {"Challenge": "Cpsc 110 assignment is too hard"}, {"Challenge": "Cpsc 404 assignment is too hard"}, {"Teamwork": "had a fight with my colleagues today"}, {"Teamwork": "had a fight with my dog today"}, {"Hackathon": "had a demo today"}, {"Hackathon": "had a demo today"}, {"Hackathon": "had a demo today"}, {"Hospital": "had a coma today"}, {"Movie": "had a movie today"}, {"Academic Concession": "had a concession today"}, {"Academic Concession": "had a concession today"}, {"Academic Concession": "had a concession today"}]]
```

Home
Calendar Diary Goal
List

Issue Board Emotion Board

There are 5 fields to choose from in the calendar:

y.yearfield,y.Yeartheme,y.Summary,yt.Theme

Example search: y.yearfield,y.Yeartheme,y.Summary,

Projection ProjectionTable

Example search: y.yearfield = 2022 AND y.Yeartheme = "Hello World"

Selection Search

havelsue: NO

number of issues by year overall

number of emotions by year overall

number of issues within the recent ? year Enter year

number of issues within ? year resolved Enter year

find all the unresolved or resolved issues that exists every year. Issue Resolved: NO

Use YearTheme

[["andy@example.com", 2021, 10], ["andy@example.com", 2022, 11], ["andy@example.com", 2023, 6],
["andy@example.com", 2024, 5]]

[Home](#)

[Calendar](#)
[List](#)
[Diary](#)
[Goal](#)

[Issue Board](#)
[Emotion Board](#)

There are 5 fields to choose from in the calendar:

y.yearfield,y.Yeartheme,y.Summary,yt.Theme

Example search: y.yearfield,y.Yeartheme,y.Summary,

Example search: y.yearfield = 2022 AND y.Yeartheme = "Hello World"

havelsue: NO

number of issues by year overall

number of emotions by year overall

number of issues within the recent ? year

number of issues within ? year resolved

find all the unresolved or resolved issues that exists every year. **Issue Resolved: NO**

Use YearTheme

```
[[{"andy@example.com", 2021, 10}, {"andy@example.com", 2022, 11}, {"andy@example.com", 2023, 6}, {"andy@example.com", 2024, 5}]]
```

[Home](#)

[Calendar](#) [List](#) [Diary](#) [Goal](#)

[Issue Board](#) [Emotion Board](#)

There are 5 fields to choose from in the calendar:

`y.yearfield,y.Yeartheme,y.Summary,yt.Theme`

Example search: `y.yearfield,y.Yeartheme,y.Summary,`

Example search: `y.yearfield = 2022 AND y.Yeartheme = "Hello World"`

havelsue: NO

number of issues by year overall

number of emotions by year overall

number of issues within the recent ? year

number of issues within ? year resolved

find all the unresolved or resolved issues that exists every year. Issue Resolved: NO

```
[[{"andy@example.com", 2021, 10}, {"andy@example.com", 2022, 11}, {"andy@example.com", 2023, 6}, {"andy@example.com", 2024, 5}]]
```

In the database, every issues is linked with an emotion as it is commonly part of writing journals. Therefore, the number of issues and emotions are the same

```
Data got from backend: [[{"email": "andy@example.com", "year": 2021, "month": 10}, {"email": "andy@example.com", "year": 2022, "month": 11}, {"email": "andy@example.com", "year": 2023, "month": 6}, {"email": "andy@example.com", "year": 2024, "month": 5}]]  
Data got from backend: [[{"email": "andy@example.com", "year": 2021, "month": 10}, {"email": "andy@example.com", "year": 2022, "month": 11}, {"email": "andy@example.com", "year": 2023, "month": 6}, {"email": "andy@example.com", "year": 2024, "month": 5}]]  
Data got from backend: [[{"email": "andy@example.com", "year": 2021, "month": 10}, {"email": "andy@example.com", "year": 2022, "month": 11}, {"email": "andy@example.com", "year": 2023, "month": 6}, {"email": "andy@example.com", "year": 2024, "month": 5}]]  
Data got from backend: [[{"email": "andy@example.com", "year": 2021, "month": 10}, {"email": "andy@example.com", "year": 2022, "month": 11}, {"email": "andy@example.com", "year": 2023, "month": 6}, {"email": "andy@example.com", "year": 2024, "month": 5}]]
```

As can be seen in here, the two codes are executed in different lines or different cases

```

359 result = json.dumps(data)
360 elif (options == "GROUPBY"):
361     verification = " u.Email = " + request.json['email'] + " "
362     # number of issues by years and within a year if specified.
363
364     if ("year" not in request.json):
365         yearcondition = ''
366     else:
367         yearcondition = " AND y.yearfield = " + str(request.json["year"])
368
369     if (request.json["table"] == "Issue"):
370         projection = 'y.yearfield as Year, Count(I.issueID) as NumberofIssue'
371         l1 = ' m.UserID = u.uID AND u.uID = y.UserID AND i.UserID = u.uID '
372         l2 = ' AND yt.theme = m.Theme '
373         l3 = ' AND yt.YearTheme = y.YearTheme AND yt.yearfield = y.yearfield AND yt.yearID = y.yearID '
374         l4 = ' AND y.yearfield = idy.yearfield AND idy.yearntheme = y.yearntheme AND idy.yearID = y.yearID '
375         l5 = ' AND EB.yearfield = y.yearfield AND EB.yearntheme = y.yearntheme AND EB.UserID = u.uID '
376         l6 = ' AND ebt.Subtypes = EB.Subtypes AND ebt EmotionID = EB.EmotionID '
377         l7 = ' AND idy.issueID = i.issueID '
378         connectingtable = l1 + l2 + l3 + l4 + l5 + l6 + l7
379         query = 'SELECT u.Email, ' + projection + ' FROM YearTheme yt, Menu m, Users u, Years y, IssueDateYear idy, EB, Issue i, EBType ebt' + ' WHERE ' + connectingtable + ' AND ' + verification + yearcondition
380         result = []
381         cursor = connection.cursor()
382         cursor.execute(query)
383         data = cursor.fetchall()
384         result = json.dumps(data)
385     # number of emotion within a year
386     elif (request.json["table"] == "EB") :
387         projection = 'y.yearfield as Year, Count(EB.EmotionID) as NumberofEmotions'
388         l1 = ' m.UserID = u.uID AND u.uID = y.UserID AND i.UserID = u.uID '
389         l2 = ' AND yt.theme = m.Theme '
390         l3 = ' AND yt.YearTheme = y.YearTheme AND yt.yearfield = y.yearfield AND yt.yearID = y.yearID '
391         l4 = ' AND y.yearfield = idy.yearfield AND idy.yearntheme = y.yearntheme AND idy.yearID = y.yearID '
392         l5 = ' AND EB.yearfield = y.yearfield AND EB.yearntheme = y.yearntheme AND EB.UserID = u.uID '
393         l6 = ' AND ebt.Subtypes = EB.Subtypes AND ebt.EmotionID = EB.EmotionID '
394         l7 = ' AND idy.issueID = i.issueID '
395         connectingtable = l1 + l2 + l3 + l4 + l5 + l6 + l7
396         query = 'SELECT u.Email, ' + projection + ' FROM YearTheme yt, Menu m, Users u, Years y, IssueDateYear idy, EB, Issue i, EBType ebt' + ' WHERE ' + connectingtable + ' AND ' + verification + yearcondition
397         result = []
398         cursor = connection.cursor()
399         cursor.execute(query)
400         data = cursor.fetchall()
401         result = json.dumps(data)

```

```

368     f "request.Json["table"] == "Issue":
369         projection = "yearfield as Year, Count(I.issueID) as NumberofIssue"
370         l1 = " u.UserID = u.UserID AND u.UserID = y.UserID AND l.UserID = u.UserID "
371         l2 = " AND y.yearTheme = m.Theme "
372         l3 = " AND y.yearTheme = y.yearTheme AND yt.yearfield = y.yearfield AND yt.yearID = y.yearID"
373         l4 = " AND y.yearfield = ly.yearfield AND ly.yeartheme = y.yeartheme AND ly.yearID = y.yearID"
374         l5 = " AND ly.yearfield = EB.yearfield AND EB.yeartheme = y.yeartheme AND EB.UserID = u.UserID "
375         l6 = " AND EB.yearfield = E.yearfield AND E.yeartheme = y.yeartheme AND E.UserID = u.UserID "
376         l7 = " AND E.yearfield = Es.yearfield AND Es.yeartheme = y.yeartheme AND Es.UserID = Es.UserID"
377         connectingtable = l1 + l2 + l3 + l4 + l5 + l6 + l7
378         query = "SELECT U.Email, " + projection + " FROM YearTheme yt, Menu m, Users u, Years y, IssueDateYear idy, EB, Issue i, EBtype ebt" + " WHERE " + connectingtable + " AND " + verification + " yearcondition " + " GROUP BY Year, u.Email";
379         result = []

```

```

384 number_of_emotions_within_a_year
385
386 if (request_json['table'] == "EB") :
387     projection_json = 'y.yearfield as Year, Count(EB.EmotionID) as NumberofEmotions'
388     projection = 'y.yearfield as Year, COUNT(u.UserID) AS UserCount'
389     connectingtable = 'y ON y.UserID = u.UserID AND u.UserID = y.UserID AND 1.UserID = u.UserID '
390     where = ' AND y.yearfield = y.yearTheme AND yt.yearfield = y.yearfield AND yt.yearID = y.yearID'
391     yearcondition = ' AND y.yearfield = idy.yearfield AND idy.yeartheme = y.yeartheme AND idy.yearID = y.yearID'
392     verification = ' AND EB.yearfield = y.yearfield AND EB.yeartheme = y.yeartheme AND EB.UserID = u.UserID '
393     ebtSubtypes = ' AND ebt.Subtypes = EB.Subtypes AND ebt.EmotionID = EB.EmotionID'
394     issuedID = ' AND idy.issuedID = 1.issuedID '
395     connectingtable = '1 + 12 + 13 + 14 + 15 + 16 + 17 '
396     query = 'SELECT U.Email, ' + projection + ' FROM YearTheme yt, Menu m, Users u, Years y, IssueDateYear idy, EB, Issue i, EBTType ebt' + ' WHERE ' + connectingtable + ' AND ' + verification + ' yearcondition ' + ' GROUP BY Year, u.Email' +
397     result = []
398     cursor = connection.cursor()

```

GROUPBY ^ Line 358-400 in server/main.py. This groups the issue or emotions by year depending on the users' choice. The query is from line 369 to line 379 for issues aggregation and from line 385 to line 395 for emotions aggregation.

Home
Calendar Diary Goal
List

Issue Board Emotion Board

There are 5 fields to choose from in the calendar:

y.yearfield,y.Yeartheme,y.Summary,yt.Theme

Example search: y.yearfield,y.Yeartheme,y.Summary,

Example search: y.yearfield = 2022 AND y.Yeartheme = "Hello World"

havelsue: NO

number of issues by year overall

number of emotions by year overall

number of issues within the recent ? year

number of issues within ? year resolved

find all the unresolved or resolved issues that exists every year. Issue Resolved: NO

Home
Calendar Diary Goal
List

Issue Board Emotion Board

There are 5 fields to choose from in the calendar:

y.yearfield,y.Yeartheme,y.Summary,yt.Theme

Example search: y.yearfield,y.Yeartheme,y.Summary,

Projection ProjectionTable

Example search: y.yearfield = 2022 AND y.Yeartheme = "Hello World"

Selection Search

havelsue: NO

number of issues by year overall

number of emotions by year overall

number of issues within the recent ? year

number of issues within ? year resolved

find all the unresolved or resolved issues that exists every year. Issue Resolved: NO

Use YearTheme

[["Fourth Year with Honours", 3, 2023], ["Second Year with Honours", 7, 2024], ["Third Year with Honours", 21, 2024]]

```

401
402
403 elif (options == 'HAVING'):
404     # number of issues within the recent * year for each theme. * to be inputted by the user
405     # use resolved or not as the step for having.
406     # DAVID: Note the name of the body field
407     verification = "u.Email = '" + request.json["email"] + "'"
408     if ("resolvedissue" not in request.json):
409         resolved = ''
410     else:
411         resolved = ' AND i.Resolved = TRUE'
412     if ("recentyearno" not in request.json):
413         maxyearforeachtheme = ''
414         recentyearcondition = ''
415     else:
416         s1 = 'WITH maxyear(Yeartheme, Maxyear) AS (SELECT yt.Yeartheme, Max(idy.yearfield) AS Maxyear'
417         s2 = ' FROM YearTheme yt, Menu m, Users u, Years y, IssueDateYear idy, EB, Issue i, EBType ebt'
418         s3 = ' WHERE m.UserID = u.uID AND u.uID = y.UserID AND i.UserID = u.uID '
419         s4 = ' AND yt.theme = m.Theme '
420         s5 = ' AND yt.YearTheme = y.YearTheme AND yt.yearfield = y.yearfield AND yt.yearID = y.yearID'
421         s6 = ' AND y.yearfield = idy.yearfield AND idy.yeartheme = y.yeartheme AND idy.yearID = y.yearID'
422         s7 = ' AND EB.yearfield = y.yearfield AND EB.yeartheme = y.yeartheme AND EB.UserID = u.uID'
423         s8 = ' AND ebt.Subtypes = EB.Subtypes AND ebt EmotionID = EB.EmotionID'
424         s9 = ' AND idy.issueID = i.issueID AND ' + verification
425         s10 = ' GROUP BY yt.Yeartheme)'
426         maxyearforeachtheme = s1 + s2 + s3 + s4 + s5 + s6 + s7 + s8 + s9 + s10
427         recentyearcondition = 'AND maxyear.Maxyear > (' + str(request.json["recentyearno"]) + ')'
428         recentyearcondition2 = 'AND maxyear.Maxyear < (' + str(request.json["recentyearno"]) + ')'
429     if ("recentyearno" in request.json and "resolvedissue" not in request.json):
430
431         l1 = ' SELECT DISTINCT y.Yeartheme as Yeartheme, Count(i.issueID) as CountIssue, maxyear.maxyear'
432         l2 = ' FROM YearTheme yt, Menu m, Users u, Years y, IssueDateYear idy, EB, Issue i, EBType ebt, maxyear'
433         l3 = ' WHERE m.UserID = u.uID AND u.uID = y.UserID AND i.UserID = u.uID '
434         l4 = ' AND yt.theme = m.Theme '
435         l5 = ' AND yt.YearTheme = y.YearTheme AND yt.yearfield = y.yearfield AND yt.yearID = y.yearID'
436         l6 = ' AND y.yearfield = idy.yearfield AND idy.yeartheme = y.yeartheme AND idy.yearID = y.yearID'
437         l7 = ' AND EB.yearfield = y.yearfield AND EB.yeartheme = y.yeartheme AND EB.UserID = u.uID'
438         l8 = ' AND ebt.Subtypes = EB.Subtypes AND ebt EmotionID = EB.EmotionID'
439         l9 = ' AND idy.issueID = i.issueID AND ' + verification
440         l10 = ' GROUP BY y.Yeartheme, maxyear.Yeartheme, maxyear.maxyear'
441         l11 = ' HAVING maxyear.Yeartheme = y.Yeartheme ' + recentyearcondition + recentyearcondition2
442     query = maxyearforeachtheme + l1 + l2 + l3 + l4 + l5 + l6 + l7 + l8 + l9 + l10 + l11

```

NESTED GROUPBY ^ line 402 to line 442 in server/main.py. The query is from line 415 to line 442 where from line 415 to line 428 is the first aggregation and the rest takes in that as a view for nested aggregation.

Home
Calendar List Diary Goal

Issue Board **Emotion Board**

There are 5 fields to choose from in the calendar:

y.yearfield,y.Yeartheme,y.Summary,yt.Theme

Example search: y.yearfield,y.Yeartheme,y.Summary,

Example search: y.yearfield = 2022 AND y.Yeartheme = "Hello World"

havelsue: NO

number of issues by year overall

number of emotions by year overall

number of issues within the recent ? year

number of issues within ? year resolved

find all the unresolved or resolved issues that exists every year. Issue Resolved: NO

Use YearTheme

[["Fourth Year with Honours", 3, 2023], ["Second Year with Honours", 7, 2024], ["Third Year with Honours", 21, 2024]]

Home
Calendar Diary Goal
List

Issue Board Emotion Board

There are 5 fields to choose from in the calendar:

y.yearfield,y.Yeartheme,y.Summary,yt.Theme

Example search: y.yearfield,y.Yeartheme,y.Summary,

Projection ProjectionTable

Example search: y.yearfield = 2022 AND y.Yeartheme = "Hello World"

Selection Search

havelsue: NO

number of issues by year overall

number of emotions by year overall

number of issues within the recent ? year Enter year

number of issues within ? year resolved 3

find all the unresolved or resolved issues that exists every year. Issue Resolved: NO

Use YearTheme

[["Fourth Year with Honours", 1, 2023], ["Third Year with Honours", 18, 2024]]

```

401
402
403     elif (options == 'HAVING'):
404         # number of issues within the recent * year for each theme. * to be inputted by the user
405         # use resolved or not as the step for having.
406         # DAVIDO: Note the name of the body field
407         verification = "u.Email = '" + request.json["email"] + "'"
408     if ("resolvedissue" not in request.json):
409         resolved = ''
410     else:
411         resolved = ' AND i.Resolved = TRUE'
412     if ("recentyearno" not in request.json):
413         maxyearforeachtheme = ''
414         recentyearcondition = ''
415     else:
416         s1 = 'WITH maxyear(Yeartheme, Maxyear) AS (SELECT yt.Yeartheme, Max(idy.yearfield) AS Maxyear'
417         s2 = ' FROM YearTheme yt, Menu m, Users u, Years y, IssueDateYear idy, EB, Issue I, EBTType ebt'
418         s3 = ' WHERE m.UserID = u.uID AND u.uID = y.UserID AND I.UserID = u.uID '
419         s4 = ' AND yt.theme = m.Theme '
420         s5 = ' AND yt.YearTheme = y.YearTheme AND yt.yearfield = y.yearfield AND yt.yearID = y.yearID'
421         s6 = ' AND y.yearfield = idy.yearfield AND idy.yeartheme = y.yeartheme AND idy.yearID = y.yearID'
422         s7 = ' AND EB.yearfield = y.yearfield AND EB.yeartheme = y.yeartheme AND EB.UserID = u.uID'
423         s8 = ' AND ebt.Subtypes = EB.Subtypes AND ebt.EmotionID = EB.EmotionID'
424         s9 = ' AND idy.issueID = I.issueID AND ' + verification
425         s10 = ' GROUP BY yt.Yeartheme)'
426         maxyearforeachtheme = s1 + s2 + s3 + s4 + s5 + s6 + s7 + s8 + s9 + s10
427         recentyearcondition = 'AND maxyear.Maxyear > (2024 - ' + str(request.json["recentyearno"]) + ')'
428         recentyearcondition2 = 'AND maxyear.Maxyear < (2024 + ' + str(request.json["recentyearno"]) + ')'
429     if ("recentyearno" in request.json and "resolvedissue" not in request.json):
430
431         l1 = ' SELECT DISTINCT y.Yeartheme as Yeartheme, Count(I.issueID) as CountIssue, maxyear.maxyear'
432         l2 = ' FROM YearTheme yt, Menu m, Users u, Years y, IssueDateYear idy, EB, Issue I, EBTType ebt, maxyear'
433         l3 = ' WHERE m.UserID = u.uID AND u.uID = y.UserID AND I.UserID = u.uID '
434         l4 = ' AND yt.theme = m.Theme '
435         l5 = ' AND yt.YearTheme = y.YearTheme AND yt.yearfield = y.yearfield AND yt.yearID = y.yearID'
436         l6 = ' AND y.yearfield = idy.yearfield AND idy.yeartheme = y.yeartheme AND idy.yearID = y.yearID'
437         l7 = ' AND EB.yearfield = y.yearfield AND EB.yeartheme = y.yeartheme AND EB.UserID = u.uID'
438         l8 = ' AND ebt.Subtypes = EB.Subtypes AND ebt.EmotionID = EB.EmotionID'
439         l9 = ' AND idy.issueID = I.issueID AND ' + verification
440         l10 = ' GROUP BY y.Yeartheme, maxyear.Yeartheme, maxyear.maxyear'
441         l11 = ' HAVING maxyear.Yeartheme = y.Yeartheme ' + recentyearcondition + recentyearcondition2
442         query = maxyearforeachtheme + l1 + l2 + l3 + l4 + l5 + l6 + l7 + l8 + l9 + l10 + l11
443     elif ("recentyearno" in request.json and "resolvedissue" in request.json):
444
445         l1 = ' SELECT DISTINCT y.Yeartheme as Yeartheme, Count(I.issueID) as CountIssue, maxyear.maxyear as maxyear'
446         l2 = ' FROM YearTheme yt, Menu m, Users u, Years y, IssueDateYear idy, EB, Issue I, EBTType ebt, maxyear'
447         l3 = ' WHERE m.UserID = u.uID AND u.uID = y.UserID AND I.UserID = u.uID '
448         l4 = ' AND yt.theme = m.Theme '
449         l5 = ' AND yt.YearTheme = y.YearTheme AND yt.yearfield = y.yearfield AND yt.yearID = y.yearID'
450         l6 = ' AND y.yearfield = idy.yearfield AND idy.yeartheme = y.yeartheme AND idy.yearID = y.yearID'
451         l7 = ' AND EB.yearfield = y.yearfield AND EB.yeartheme = y.yeartheme AND EB.UserID = u.uID'
452         l8 = ' AND ebt.Subtypes = EB.Subtypes AND ebt.EmotionID = EB.EmotionID'
453         l9 = ' AND idy.issueID = I.issueID ' + resolved + ' AND ' + verification
454         l10 = ' GROUP BY y.Yeartheme, maxyear.Yeartheme, maxyear.maxyear'
455         l11 = ' HAVING maxyear.Yeartheme = y.Yeartheme ' + recentyearcondition + recentyearcondition2
456         query = maxyearforeachtheme + l1 + l2 + l3 + l4 + l5 + l6 + l7 + l8 + l9 + l10 + l11

```

HAVING ^ line 401 to line 456 and the query is from line 443 to line 456. The query is from line 443 to line 456 which may take in the view from an aggregation in line 415 to line 429 to check on the resolved issues along with the recent year condition in the having clause.

[Home](#)

[Calendar](#)
[List](#)
[Diary](#)
[Goal](#)

[Issue Board](#)
[Emotion Board](#)

There are 5 fields to choose from in the calendar:

y.yearfield,y.Yeartheme,y.Summary,yt.Theme

Example search: y.yearfield,y.Yeartheme,y.Summary,

Example search: y.yearfield = 2022 AND y.Yeartheme = "Hello World"

havelissue: NO

number of issues by year overall

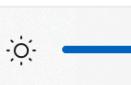
number of emotions by year overall

number of issues within the recent ? year

number of issues within ? year resolved

find all the unresolved or resolved issues that exists every year.

[[{"Fourth Year with Honours", 1, 2023}, {"Third Year with Honours", 18, 2024}]]



[Home](#)

[Calendar](#) [Diary](#) [Goal](#)

[List](#)

[Issue Board](#) [Emotion Board](#)

There are 5 fields to choose from in the calendar:

y.yearfield,y.Yeartheme,y.Summary,yt.Theme

Example search: y.yearfield,y.Yeartheme,y.Summary,

Example search: y.yearfield = 2022 AND y.Yeartheme = "Hello World"

havelsue: NO

number of issues by year overall

number of emotions by year overall

number of issues within the recent ? year

number of issues within ? year resolved

find all the unresolved or resolved issues that exists every year. **Issue Resolved: YES**

Use YearField

```
[[{"Teamwork", 2024, "andy@example.com", "691e766a-714c-4512-a650-ecf17b40227d"}, {"Teamwork", 2023, "andy@example.com", "691e766a-714c-4512-a650-ecf17b40227d"}, {"Teamwork", 2021, "andy@example.com", "691e766a-714c-4512-a650-ecf17b40227d"}, {"Teamwork", 2022, "andy@example.com", "691e766a-714c-4512-a650-ecf17b40227d"}, {"Hackathon", 2021, "andy@example.com", "691e766a-714c-4512-a650-ecf17b40227d"}, {"Hackathon", 2022, "andy@example.com", "691e766a-714c-4512-a650-ecf17b40227d"}, {"Hackathon", 2023, "andy@example.com", "691e766a-714c-4512-a650-ecf17b40227d"}, {"Hackathon", 2024, "andy@example.com", "691e766a-714c-4512-a650-ecf17b40227d}]]
```

Here we are outputting all the issues with the years so that we can check that it is the right one

Home
Calendar Diary Goal
List

Issue Board Emotion Board

There are 5 fields to choose from in the calendar:

y.yearfield,y.Yeartheme,y.Summary,yt.Theme

Example search: y.yearfield,y.Yeartheme,y.Summary,

Projection ProjectionTable

Example search: y.yearfield = 2022 AND y.Yeartheme = "Hello World"

Selection Search

havelsissue: NO

number of issues by year overall

number of emotions by year overall

number of issues within the recent ? year Enter year

number of issues within ? year resolved Enter year

find all the unresolved or resolved issues that exists every year. Issue Resolved: NO

Use YearField

```
[["Teamwork", 2024, "andy@example.com", "691e766a-714c-4512-a650-ecf17b40227d"],  
 ["Teamwork", 2023, "andy@example.com", "691e766a-714c-4512-a650-ecf17b40227d"],  
 ["Teamwork", 2021, "andy@example.com", "691e766a-714c-4512-a650-ecf17b40227d"],  
 ["Teamwork", 2022, "andy@example.com", "691e766a-714c-4512-a650-ecf17b40227d"],  
 ["Hackathon", 2021, "andy@example.com", "691e766a-714c-4512-a650-ecf17b40227d"],  
 ["Hackathon", 2022, "andy@example.com", "691e766a-714c-4512-a650-ecf17b40227d"],  
 ["Hackathon", 2023, "andy@example.com", "691e766a-714c-4512-a650-ecf17b40227d"],  
 ["Hackathon", 2024, "andy@example.com", "691e766a-714c-4512-a650-ecf17b40227d"]]
```

Home
Calendar Diary Goal
List

Issue Board Emotion Board

There are 5 fields to choose from in the calendar:

y.yearfield,y.Yeartheme,y.Summary,yt.Theme

Example search: y.yearfield,y.Yeartheme,y.Summary,

Projection ProjectionTable

Example search: y.yearfield = 2022 AND y.Yeartheme = "Hello World"

Selection Search

havelsue: NO

number of issues by year overall

number of emotions by year overall

number of issues within the recent ? year

Enter year

number of issues within ? year resolved

Enter year

find all the unresolved or resolved issues that exists every year. Issue Resolved: NO

Use YearField

```
[["Academic Concession", 2021, "andy@example.com", "691e766a-714c-4512-a650-ecf17b40227d"],  
 ["Academic Concession", 2022, "andy@example.com", "691e766a-714c-4512-a650-ecf17b40227d"],  
 ["Academic Concession", 2023, "andy@example.com", "691e766a-714c-4512-a650-ecf17b40227d"],  
 ["Academic Concession", 2024, "andy@example.com", "691e766a-714c-4512-a650-ecf17b40227d"],  
 ["Challenge", 2023, "andy@example.com", "691e766a-714c-4512-a650-ecf17b40227d"], ["Challenge",  
 2022, "andy@example.com", "691e766a-714c-4512-a650-ecf17b40227d"], ["Challenge", 2021,  
 "andy@example.com", "691e766a-714c-4512-a650-ecf17b40227d"], ["Challenge", 2024,  
 "andy@example.com", "691e766a-714c-4512-a650-ecf17b40227d"]]
```

Home
Calendar Diary Goal
List

Issue Board **Emotion Board**

There are 5 fields to choose from in the calendar:

y.yearfield,y.Yeartheme,y.Summary,yt.Theme

Example search: y.yearfield,y.Yeartheme,y.Summary,

Example search: y.yearfield = 2022 AND y.Yeartheme = "Hello World"

havelissue: NO

number of issues by year overall

number of emotions by year overall

number of issues within the recent ? year

number of issues within ? year resolved

find all the unresolved or resolved issues that exists every year.

Issue Resolved: NO

Use YearTheme

```
[[{"Academic Concession", 2021, "andy@example.com", "691e766a-714c-4512-a650-ecf17b40227d"},  
 {"Academic Concession", 2022, "andy@example.com", "691e766a-714c-4512-a650-ecf17b40227d"},  
 {"Academic Concession", 2023, "andy@example.com", "691e766a-714c-4512-a650-ecf17b40227d"},  
 {"Academic Concession", 2024, "andy@example.com", "691e766a-714c-4512-a650-ecf17b40227d"},  
 {"Challenge", 2023, "andy@example.com", "691e766a-714c-4512-a650-ecf17b40227d"}, ["Challenge", 2021,  
 2022, "andy@example.com", "691e766a-714c-4512-a650-ecf17b40227d"], ["Challenge", 2021,  
 "andy@example.com", "691e766a-714c-4512-a650-ecf17b40227d"], ["Challenge", 2024,  
 "andy@example.com", "691e766a-714c-4512-a650-ecf17b40227d"]]]
```

Home
Calendar Diary Goal
List

Issue Board **Emotion Board**

There are 5 fields to choose from in the calendar:

y.yearfield,y.Yeartheme,y.Summary,yt.Theme

Example search: y.yearfield,y.Yeartheme,y.Summary,

Example search: y.yearfield = 2022 AND y.Yeartheme = "Hello World"

havelsue: NO

number of issues by year overall

number of emotions by year overall

number of issues within the recent ? year

number of issues within ? year resolved

find all the unresolved or resolved issues that exists every year. Issue Resolved: NO

Use YearTheme

```
[[{"Academic Concession", "First Year with Honours", "andy@example.com", "691e766a-714c-4512-a650-ecf17b40227d"}, {"Academic Concession", "First Year with Honours", "andy@example.com", "691e766a-714c-4512-a650-ecf17b40227d"}, {"Academic Concession", "First Year with Honours", "andy@example.com", "691e766a-714c-4512-a650-ecf17b40227d"}, {"Academic Concession", "First Year with Honours", "andy@example.com", "691e766a-714c-4512-a650-ecf17b40227d"}]]
```

Home
Calendar Diary Goal
List

Issue Board **Emotion Board**

There are 5 fields to choose from in the calendar:

y.yearfield,y.Yeartheme,y.Summary,yt.Theme

Example search: y.yearfield,y.Yeartheme,y.Summary,

Example search: y.yearfield = 2022 AND y.Yeartheme = "Hello World"

havelsIssue: NO

number of issues by year overall

number of emotions by year overall

number of issues within the recent ? year

number of issues within ? year resolved

find all the unresolved or resolved issues that exists every year.

Use YearTheme

```
[[{"Academic Concession", "First Year with Honours", "andy@example.com", "691e766a-714c-4512-a650-ecf17b40227d"}, {"Academic Concession", "First Year with Honours", "andy@example.com", "691e766a-714c-4512-a650-ecf17b40227d"}, {"Academic Concession", "First Year with Honours", "andy@example.com", "691e766a-714c-4512-a650-ecf17b40227d"}, {"Academic Concession", "First Year with Honours", "andy@example.com", "691e766a-714c-4512-a650-ecf17b40227d"}]]
```

[Home](#)

[Calendar](#)

[Diary](#)

[Goal](#)

[List](#)

[Issue Board](#)

[Emotion Board](#)

There are 5 fields to choose from in the calendar:

y.yearfield,y.Yeartheme,y.Summary,yt.Theme

Example search: y.yearfield,y.Yeartheme,y.Summary,

Example search: y.yearfield = 2022 AND y.Yeartheme = "Hello World"

havelsue: NO

number of issues by year overall

number of emotions by year overall

number of issues within the recent ? year
Enter year

number of issues within ? year resolved
Enter year

find all the unresolved or resolved issues that exists every year.
Issue Resolved: YES

```

466     result = main.mainDump(options)
467     if (options == 'division'):
468         # find all the issues that have not been resolved overall that exists in every year.
469         verification = " u.Email = " + request.json["email"] + " "
470         if ("attribute" in request.json):
471             if ("isissuesolved" in request.json):
472                 issuecondition = request.json["isissuesolved"]
473                 11 = " SELECT l.issueName , 1.IssuerName , y.' + str(request.json["attribute"]).replace(' ', '') + ' , u.Email , u.uID"
474                 12 = " FROM Users u , Issue 1 , YearTheme yt , Menu m , Years y , IssueDateYear idy"
475                 13 = " WHERE u.Email = " + str(request.json["email"]) + " AND u.uID = 1.UserID AND"
476                 14 = " 1.UserID = u.uID AND u.uID = y.UserID AND 1.UserID = u.uID "
477                 15 = " AND yt.YearTheme = m.Theme AND yt.MenuID = m.MenuID AND yt.yearID = yt.yearID "
478                 16 = " AND yt.yearField = y.yearField AND idy.yearField = y.yearField"
479                 17 = " AND idy.yearField = idy.yearField AND idy.yearTheme = y.yearTheme AND idy.yearID = y.yearID"
480                 18 = " AND idy.issueID = 1.issueID "
481                 19 = " AND NOT EXISTS (SELECT DISTINCT y2." + str(request.json["attribute"]).replace(' ', '') + " FROM Years y3 , Issue 12"
482                 110 = " WHERE idy.uID = y2.uID AND idy.yearID = y2.yearID EXCEPT (SELECT "
483                 111 = " idy2." + str(request.json["attribute"]).replace(' ', '') + " FROM Issue 13 , IssuedateYear idy2 , Years y4 WHERE idy3.Resolved = " + str(issuecondition).capitalize()
484                 112 = " AND idy3.IssuerName = 1.IssuerName AND idy3.uID = idy2.uID AND idy2.issueID = idy3.issueID AND y4.yearID = idy2.yearID AND y4.yearField = idy2.yearField AND y4.yearTheme = idy2.yearTheme AND u.uID = y4.UserID AND idy3.UserID = u.uID )"
485                 query = 11 + 12 + 13 + 14 + 15 + 16 + 17 + 18 + 19 + 110 + 111 + 112
486             else:
487                 return jsonify({
488                     "error": "No issue keys for checking condition (DIVISION)"
489                 })

```

```

466     if ("attribute" in request.json):
467         ("isissuesolved" in request.json):
468             issuecondition = request.json["isissuesolved"]
469             11 = " SELECT l.issueName as IssuerName , y.' + str(request.json["attribute"]).replace(' ', '') + ' , u.Email , u.uID"
470             12 = " FROM Users u , Issue 1 , YearTheme yt , Menu m , Years y , IssueDateYear idy"
471             13 = " WHERE u.Email = " + str(request.json["email"]) + " AND u.uID = 1.UserID AND"
472             14 = " 1.UserID = u.uID AND u.uID = y.UserID AND idy.yearID = y.yearID"
473             15 = " AND yt.Theme = y.Theme AND yt.MenuID = m.MenuID AND yt.yearID = yt.yearID "
474             16 = " AND yt.YearTheme = y.YearTheme AND yt.yearField = y.yearField AND yt.yearID = y.yearID"
475             17 = " AND yt.yearField = idy.yearField AND idy.yearTheme = y.yearTheme AND idy.yearID = y.yearID"
476             18 = " AND idy.issueID = 1.issueID "
477             19 = " AND NOT EXISTS (SELECT DISTINCT y3." + str(request.json["attribute"]).replace(' ', '') + " FROM Years y3 , Issue 12"
478             110 = " WHERE idy.uID = y3.uID AND idy.yearID = y3.yearID EXCEPT (SELECT "
479             111 = " idy2." + str(request.json["attribute"]).replace(' ', '') + " FROM Issue 13 , IssuedateYear idy2 , Years y4 WHERE idy3.Resolved = " + str(issuecondition).capitalize()
480             112 = " AND idy3.IssuerName = 1.IssuerName AND idy3.uID = idy2.uID AND idy2.issueID = idy3.issueID AND y4.yearID = idy2.yearID AND y4.yearField = idy2.yearField AND y4.yearTheme = idy2.yearTheme AND u.uID = y4.UserID AND idy3.UserID = u.uID )"
481             query = 11 + 12 + 13 + 14 + 15 + 16 + 17 + 18 + 19 + 110 + 111 + 112

```

DIVISION ^ line 466 to 485 in server/main.py; query is from line 472 to line 484.

Home
Calendar Diary Goal
List

Add Diary

diarytheme

What part of diary do you want to see? Diarytheme or Menutheme?

If diarytheme, write in diarytheme else menutheme

searchWithCondition

Home
Calendar Diary Goal
List

Add Diary

diarytheme

What part of diary do you want to see? Diarytheme or Menutheme?

If diarytheme, write in diarytheme else menutheme

searchWithCondition [{"Health and Fitness", "Social",

"andy@example.com", "Andy"}, {"Health and Fitness", "Health", "andy@example.com", "Andy"},
["Academics", "Work", "andy@example.com", "Andy"]]

Home
Calendar List Diary Goal

Add Diary

menutheme

What part of diary do you want to see? Diarytheme or Menutheme?

If diarytheme, write in diarytheme else menutheme

 searchWithCondition [{"Health and Fitness", "Social", "andy@example.com", "Andy"}, {"Health and Fitness", "Health", "andy@example.com", "Andy"}, {"Academics", "Work", "andy@example.com", "Andy"}]

"andy@example.com", "Andy"], ["Health and Fitness", "Health", "andy@example.com", "Andy"], ["Academics", "Work", "andy@example.com", "Andy"]]

Home
Calendar Diary Goal
List

Add Diary

menutheme

What part of diary do you want to see? Diarytheme or Menutheme?

If diarytheme, write in diarytheme else menutheme

searchWithCondition [[{"Social": "Health and Fitness",

"andy@example.com", "Andy"], [{"Health": "Health and Fitness", "andy@example.com": "Andy"},
 {"Work": "Academics", "andy@example.com": "Andy"}]]



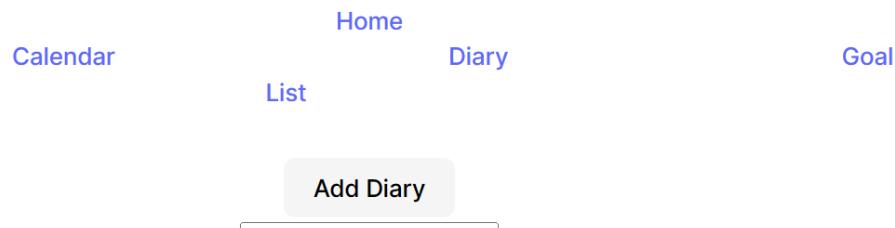
The above textbox is for inputting the field on what part of the diary you will want to see. What part of diary do you want to see? Diarytheme or Menutheme?

Eitherway, you will receive the result being a list of items each wrapped in square brackets. If diarytheme, write in diarytheme else menutheme. For diarytheme, for each item of the result you get back, it will be in order of menutheme, diarytheme, email and user's name

For menutheme, for each item of the result you get back, it will be in order of diarytheme, menutheme, email and user's name

For the condition, you can choose which results with only menutheme of certain name or diarytheme with certain name. For diarytheme condition the format is: d.DiaryTheme = '\name\' For menutheme condition the format is: dt.theme = '\name\' You can add AND in between the conditions to find a specific thing.

```
d.DiaryTheme = 'Social'      searchWithCondition  [[{"Health and Fitness", "Social",  
"andy@example.com", "Andy"}, [{"Health and Fitness", "Health", "andy@example.com", "Andy"},  
["Academics", "Work", "andy@example.com", "Andy"]]
```



The above textbox is for inputting the field on what part of the diary you will want to see. What part of diary do you want to see? Diarytheme or Menutheme?

Eitherway, you will receive the result being a list of items each wrapped in square brackets. If diarytheme, write in diarytheme else menutheme. For diarytheme, for each item of the result you get back, it will be in order of menutheme, diarytheme, email and user's name

For menutheme, for each item of the result you get back, it will be in order of diarytheme, menutheme, email and user's name

For the condition, you can choose which results with only menutheme of certain name or diarytheme with certain name. For diarytheme condition the format is: d.DiaryTheme = '\name\' For menutheme condition the format is: dt.theme = '\name\' You can add AND in between the conditions to find a specific thing.

```
d.DiaryTheme = 'Social'      searchWithCondition  [[["Health and Fitness", "Social",
"andy@example.com", "Andy"]]]
```



The above textbox is for inputting the field on what part of the diary you will want to see. What part of diary do you want to see? Diarytheme or Menutheme?

Eitherway, you will receive the result being a list of items each wrapped in square brackets. If diarytheme, write in diarytheme else menutheme. For diarytheme, for each item of the result you get back, it will be in order of menutheme, diarytheme, email and user's name

For menutheme, for each item of the result you get back, it will be in order of diarytheme, menutheme, email and user's name

For the condition, you can choose which results with only menutheme of certain name or diarytheme with certain name. For diarytheme condition the format is: d.DiaryTheme = '\name\' For menutheme condition the format is: dt.theme = '\name\' You can add AND in between the conditions to find a specific thing.

```

dt.Theme = 'Health and Fitn
searchWithCondition  [[["Health and Fitness", "Social",
"andy@example.com", "Andy"]]
```



`diarytheme`

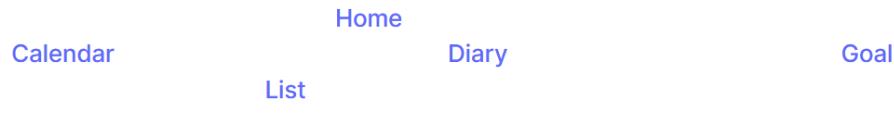
The above textbox is for inputting the field on what part of the diary you will want to see. What part of diary do you want to see? Diarytheme or Menutheme?

Eitherway, you will receive the result being a list of items each wrapped in square brackets. If diarytheme, write in diarytheme else menutheme. For diarytheme, for each item of the result you get back, it will be in order of menutheme, diarytheme, email and user's name

For menutheme, for each item of the result you get back, it will be in order of diarytheme, menutheme, email and user's name

For the condition, you can choose which results with only menutheme of certain name or diarytheme with certain name. For diarytheme condition the format is: `d.DiaryTheme = '\name'` For menutheme condition the format is: `dt.theme = '\name'` You can add AND in between the conditions to find a specific thing.

```
dt.Theme = 'Health and Fitn[ searchWithCondition ["Health and Fitness", "Health",  
"andy@example.com", "Andy"], ["Health and Fitness", "Social", "andy@example.com", "Andy"]]
```



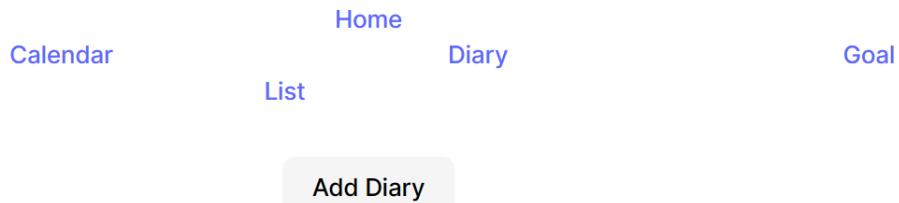
The above textbox is for inputting the field on what part of the diary you will want to see. What part of diary do you want to see? Diarytheme or Menutheme?

Eitherway, you will receive the result being a list of items each wrapped in square brackets. If diarytheme, write in diarytheme else menutheme. For diarytheme, for each item of the result you get back, it will be in order of menutheme, diarytheme, email and user's name

For menutheme, for each item of the result you get back, it will be in order of diarytheme, menutheme, email and user's name

For the condition, you can choose which results with only menutheme of certain name or diarytheme with certain name. For diarytheme condition the format is: d.DiaryTheme = '\name\' For menutheme condition the format is: dt.theme = '\name\' You can add AND in between the conditions to find a specific thing.

```
ND d.DiaryTheme = 'Health'    searchWithCondition  [[{"Health and Fitness", "Health",  
"andy@example.com", "Andy"}, [{"Health and Fitness", "Social", "andy@example.com", "Andy"}]]
```



The above textbox is for inputting the field on what part of the diary you will want to see. What part of diary do you want to see? Diarytheme or Menutheme?

Eitherway, you will receive the result being a list of items each wrapped in square brackets. If diarytheme, write in diarytheme else menutheme. For diarytheme, for each item of the result you get back, it will be in order of menutheme, diarytheme, email and user's name

For menutheme, for each item of the result you get back, it will be in order of diarytheme, menutheme, email and user's name

For the condition, you can choose which results with only menutheme of certain name or diarytheme with certain name. For diarytheme condition the format is: d.DiaryTheme = '\name\' For menutheme condition the format is: dt.theme = '\name\' You can add AND in between the conditions to find a specific thing.

```
dt.Theme = 'Health and Fitn[  
    searchWithCondition  [[["Health and Fitness", "Health",  
        "andy@example.com", "Andy"]]
```

```

526     elif ("request.method == \"POST\" and (task == \"render\") :"
527         # Show all users diaries
528         if ("condition" in request.json): # to be applied with menutheme or diarytheme
529             condition = request.json["condition"]
530         else :
531             condition = ""
532         if ("field" in request.json):
533             field = request.json["field"]
534             if (str(field).lower() == "diarytheme"):
535                 connectingboards = "d.UserID = u.userID AND dt.Diarytheme = d.diarytheme AND dt.MenuID = m.MenuID AND m.UserID = u.userID"
536                 verification = "u.Email = " + request.json["email"] + ""
537                 query = "SELECT dt.theme as menuTheme, d.DiaryTheme as DiaryTheme, u.Email, u.UserName FROM Diary d, DiaryTheme dt, Users u, Menu m WHERE" + connectingboards + " AND " + verification + " AND " + condition + ";"
538                 cursor = connection.cursor()
539                 cursor.execute(query)
540             elif (str(field).lower() == "menutheme"):
541                 connectingboards = "d.UserID = u.userID AND dt.Diarytheme = d.diarytheme AND dt.MenuID = m.MenuID AND m.UserID = u.userID"
542                 verification = "u.Email = " + request.json["email"] + ""
543                 query = "SELECT d.DiaryTheme as DiaryTheme, dt.theme as menuTheme, u.Email, u.UserName FROM Diary d, DiaryTheme dt, Users u, Menu m WHERE" + connectingboards + " AND " + verification + " AND " + condition + ";"
544                 cursor = connection.cursor()
545                 cursor.execute(query)
546             else:
547                 return jsonify({
548                     "error": "The field is inappropriate"
549                 })
550             data = cursor.fetchall()
551             result = json.dumps(data)
552             connection.close()
553         else :
554             return jsonify({
555                 "error": "No field given for render"
556             })
557         return jsonify({
558             "result": result
559         })
560

```

query:

```

532     if ("field" in request.json):
533         field = request.json["field"]
534         if (str(field).lower() == "diarytheme"):
535             connectingboards = "d.UserID = u.userID AND dt.Diarytheme = d.diarytheme AND dt.MenuID = m.MenuID AND m.UserID = u.userID"
536             verification = "u.Email = " + request.json["email"] + ""
537             query = "SELECT dt.theme as menuTheme, d.DiaryTheme as DiaryTheme, u.Email, u.UserName FROM Diary d, DiaryTheme dt, Users u, Menu m WHERE" + connectingboards + " AND " + verification + " AND " + condition + ";"
538             cursor = connection.cursor()
539             cursor.execute(query)
540         elif (str(field).lower() == "menutheme"):
541             connectingboards = "d.UserID = u.userID AND dt.Diarytheme = d.diarytheme AND dt.MenuID = m.MenuID AND m.UserID = u.userID"
542             verification = "u.Email = " + request.json["email"] + ""
543             query = "SELECT d.DiaryTheme as DiaryTheme, dt.theme as menuTheme, u.Email, u.UserName FROM Diary d, DiaryTheme dt, Users u, Menu m WHERE" + connectingboards + " AND " + verification + " AND " + condition + ";"
544             cursor = connection.cursor()
545             cursor.execute(query)
546         else:
547             return jsonify({
548                 "error": "The field is inappropriate"
549             })
550         data = cursor.fetchall()

```

JOIN ^ line 526 to line 560 and the query is from line 532 to line 544