SERVICE REQUESTS DOCUMENTATION:

all requests return a json with the fields ['success'] (boolean, true if succeeded, false if it didnt) and a ['message'] field, which contains the error message if the database operation didn't succeed with the http code 500, or the result of the request and the http code 200 if it did succeed.

1. For the user service:

1. for registering a REGULAR user:

POST request to **http://localhost/web/proiect/services/UserService/register** with a json payload containing the fields ['username'], ['email'] and ['password'] (the password must not be sent hashed);

if success, returns:

```
{
   "success": true,
   "message": "User created successfully"
}
```

2. for loging in the user

POST request to **http://localhost/web/proiect/services/UserService/login** with a json payload containing the email and password

if success, returns:

```
{
   "success": true,
   "message": "Login successful"
}
```

3. for getting the user's details by id

GET request to **http://localhost/web/proiect/services/UserService/user/{id}**(the id will be accordingly set in the url when sending the request) ;

if success, returns the user's info from the database like:

```
{
   "success": true,
   "message": {
      "user_id": 1,
      "username": "updateduser",
      "email": "updateduser@example.com",
      "password":
"$2y$10$R7ZnBpCdLn7QPvtMsHf1Wes\/pQDoH8VmtTPAM\/bk1INvyen31n8o2",
      "remember_me_token": "vdvd",
      "reset_password_token": "vdvd",
      "reset_password_token_expires": null,
      "change_email_token": null,
      "change_email_token_expires": null,
      "delete_account_token": null,
      "delete_account_token_expires": null,
      "role": "regular"
   }
}
```

4. for getting the user's details by email

GET request to **http://localhost/web/proiect/services/UserService/user/{email}**(the email will be accordingly set in the url when sending the request) ;

if success, returns the user's info from the database like:

```
{
    "success": true,
    "message": {
        "user_id": 1,
        "username": "updateduser",
        "email": "updateduser@example.com",
        "password":
"$2y$10$R7ZnBpCdLn7QPvtMsHf1Wes\/pQDoH8VmtTPAM\/bk1INvyen31n8o2",
        "remember_me_token": "vdvd",
        "reset_password_token": "vdvd",
        "reset_password_token_expires": null,
        "change_email_token": null,
        "change_email_token_expires": null,
        "delete_account_token": null,
        "delete_account_token_expires": null,
        "role": "regular"
    }
}
```

5. for getting all of the **regular** users' details from the database

GET request to **http://localhost/web/proiect/services/UserService/users**
in case of success, the ['message'] field will contain all of the users from the database and their info as a json

Example:

```
{
    "success": true,
    "message": [
        {
            "user_id": 1,
            "username": "updateduser",
            "email": "updateduser@example.com",
            "password":
"$2y$10$R7ZnBpCdLn7QPvtMsHf1Wes\/pQDoH8VmtTPAM\/bk1INvyen31n8o2",
            "remember_me_token": "vdvd",
            "reset_password_token": "vdvd",
            "reset_password_token_expires": null,
            "change_email_token": null,
            "change_email_token_expires": null,
            "delete_account_token": null,
            "delete_account_token_expires": null,
            "role": "regular"
        },
        {
            "user_id": 3,
            "username": "newuser2",
            "email": "newuser2@example.com",
            "password":
"$2y$10$Xm9NGVAIwaEqooUcohFLQOLpkuqcAMdc2Hp5dqTfAAFy13Tw.ZHMS",
            "remember_me_token": null,
            "reset_password_token": null,
            "reset_password_token_expires": null,
            "change_email_token": null,
            "change_email_token_expires": null,
            "delete_account_token": null,
            "delete_account_token_expires": null,
            "role": "regular"
        }
    ]
}
```

6. for updating a users info in the database

PUT request to **http://localhost/web/proiect/services/UserService/user/{id}** containing a json payload with the fields you want to update and their values.

Example of an array (will be converted to json before sending it):

```
$userData = [
    'username' => 'newuser',
    'email' => 'newuser@example.com',
    'password' => 'password123'
];
```

if succeeded, it will return:
```
{
   "success": true,
   "message": "User updated successfully"
}
```

7. for deleting a user from the database by id

DELETE request to **http://localhost/web/proiect/services/UserService/user/{id}** (the id will be accordingly set in the url when sending the request)

if succeeded, it will return:
```
{
   "success": true,
   "message": "User deleted successfully"
}
```

2. For the form service

1. for adding a form:

POST request to **http://localhost/web/proiect/services/FormService/form** with a json payload containing the fields [‘user_id’], [‘title’], [‘description’], [‘feedback_type’], [‘is_published’], [‘answer_time’]

if success, returns:
```
{
   "success": true,
   "message": "'Form created successfully"
   “form_id” : { id }
}
```

2. for getting the form’s details by id

GET request to **http://localhost/web/proiect/services/FormService/form/{id}**(the id will be accordingly set in the url when sending the request) ;

if success, returns:
```
{
   "success": true,
   "message": {
      "form_id": 6,
      "user_id": 3,
      "reported": 1,
      "title": "titlu",
      "description": "descriere",
      "is_published": 1,
      "created_at": "2024-06-25 01:01:37",
      "feedback_type": "comments",
      "answer_time": "2024-07-03 04:03:00",
      "file_path": "6679ecc1566ff.jpg"
   }
}
```

3. for getting the public forms

GET request to **http://localhost/web/proiect/services/FormService/public-forms**

if success, returns:

```
{
    "success": true,
    "message": [
        {
            "form_id": 6,
            "user_id": 3,
            "reported": 1,
            "title": "titlu",
            "description": "descriere",
            "is_published": 1,
            "created_at": "2024-06-25 01:01:37",
            "feedback_type": "comments",
            "answer_time": "2024-07-03 04:03:00",
            "file_path": "6679ecc1566ff.jpg"
        },
        {
            "form_id": 5,
            "user_id": 3,
            "reported": 0,
            "title": "titlu",
            "description": "descriere",
            "is_published": 1,
            "created_at": "2024-06-24 21:39:16",
            "feedback_type": "comments",
            "answer_time": "2024-06-26 21:43:00",
            "file_path": "6679bd544255c.png"
        }
    ]
}
```

4. for getting the reported forms

GET request to **http://localhost/web/proiect/services/FormService/reported-forms** if success, returns:

```
{
   "success": true,
   "message": [
      {
         "form_id": 6,
         "user_id": 3,
         "reported": 1,
         "title": "titlu",
         "description": "descriere",
         "is_published": 1,
         "created_at": "2024-06-25 01:01:37",
         "feedback_type": "comments",
         "answer_time": "2024-07-03 04:03:00",
         "file_path": "6679ecc1566ff.jpg"
      }
   ]
}
```

5. for getting all of the forms from the database

GET request to **http://localhost/web/proiect/services/FormService/forms**

in case of success, the [‘message’] field will contain all of the forms from the database and their info as a json

Example:

```
{
   "success": true,
   "message": [
      {
         "form_id": 7,
         "user_id": 3,
         "reported": 0,
         "title": "form privat",
         "description": "descriere",
         "is_published": 0,
         "created_at": "2024-06-25 01:03:04",
         "feedback_type": "suggestions",
         "answer_time": "2024-07-04 01:02:00",
         "file_path": "6679ed18419ab.png"
      },
      {
         "form_id": 6,
         "user_id": 3,
         "reported": 1,
         "title": "titlu",
         "description": "descriere",
         "is_published": 1,
         "created_at": "2024-06-25 01:01:37",
         "feedback_type": "comments",
         "answer_time": "2024-07-03 04:03:00",
         "file_path": "6679ecc1566ff.jpg"
      }
   ]
}
```

6. for updating a form's 'report' status info in the database

PUT request to **http://localhost/web/proiect/services/FormService/report-form/{id}** or **http://localhost/web/proiect/services/FormService/cancel-report-form/{id}**

if succeeded, it will return:

```
{
    "success": true,
    "message": "Form reported successfully"
}
```

or

```
{
    "success": true,
    "message": "Cancel report was successful"
}
```

7. for deleting a form from the database by id

DELETE request to **http://localhost/web/proiect/services/FormService/form/{id}** (the id will be accordingly set in the url when sending the request)

if succeeded, it will return:

```
{
    "success": true,
    "message": "Form and associated file deleted successfully"
}
```

8. for deleting all of the forms created by a user, by the user's id

DELETE request to
http://localhost/web/proiect/services/FormService/users-forms/{id} (the id will be accordingly set in the url when sending the request)

if succeeded, it will return:

```
{
    "success": true,
    "message": "Forms and associated files deleted successfully"
}
```

3. For the Answer Service

1. for adding an answer to a form:

POST request to **http://localhost/web/proiect/services/AnswerService/answer** with a json payload containing the fields [‘form_id’], [‘content’], [users_age’], [‘emotion_type’], [‘gender’], [‘education_level’], [‘experience’]

if success, returns:

```
{
    "success": true,
    "message": "Answer added successfully!"
}
```

2. for getting the answers of a form by the form’s id

GET request to **http://localhost/web/proiect/services/AnswerService/answers/{id}**(the id will be accordingly set in the url when sending the request) ;

if success, returns:

```
{
    "success": true,
    "message": [
        {
            "answer_id": 8,
            "form_id": 6,
            "users_age": 23,
            "content": "dfg",
            "emotion_type": "Trust",
            "created_at": "2024-06-25 01:02:00",
            "gender": "female",
            "education_level": "high_school",
            "experience": "weekly"
        }
    ]
}
```

3. for getting the form’s statistics

GET request to
http://localhost/web/proiect/services/AnswerService/get-statistics/{id}(the id will be accordingly set in the url when sending the request) ;

if success, returns:

```
{
   "success": true,
   "message": {
      "emotion_distribution": [
         {
            "emotion_type": "Trust",
            "count": 1
         }
      ],
      "age_distribution": [
         {
            "users_age": 23,
            "count": 1
         }
      ],
      "gender_distribution": [
         {
            "gender": "female",
            "count": 1
         }
      ],
      "education_distribution": [
         {
            "education_level": "high_school",
            "count": 1
         }
      ],
      "experience_distribution": [
         {
            "experience": "weekly",
            "count": 1
         }
      ],
      "total_answers": 1,
      "average_age": "23.0000"
   }
}
```

THE DATABASES

```
CREATE DATABASE IF NOT EXISTS foe_app;
USE foe_app;
DROP TABLE IF EXISTS answers;
DROP TABLE IF EXISTS forms;
DROP TABLE IF EXISTS users;

CREATE TABLE IF NOT EXISTS users (
    user_id INT AUTO_INCREMENT PRIMARY KEY,
    username VARCHAR(255) NOT NULL,
    email VARCHAR(255) UNIQUE NOT NULL,
    password VARCHAR(255) NOT NULL,
    remember_me_token VARCHAR(255) UNIQUE,
    reset_password_token VARCHAR(255) UNIQUE,
    reset_password_token_expires DATETIME,
    change_email_token VARCHAR(255) UNIQUE,
    change_email_token_expires DATETIME,
    delete_account_token VARCHAR(255) UNIQUE,
    delete_account_token_expires DATETIME,
    role ENUM('regular', 'admin') NOT NULL DEFAULT 'regular'
);

CREATE TABLE IF NOT EXISTS forms (
    form_id INT AUTO_INCREMENT PRIMARY KEY,
    user_id INT NOT NULL,
    reported BOOLEAN NOT NULL DEFAULT FALSE,
    title VARCHAR(100) NOT NULL,
    description VARCHAR(100),
    is_published BOOLEAN NOT NULL DEFAULT FALSE,
    created_at DATETIME NOT NULL DEFAULT CURRENT_TIMESTAMP,
    feedback_type ENUM('comments', 'suggestions', 'questions') NOT NULL DEFAULT
'comments',
    answer_time DATETIME,
    file_path VARCHAR(255),
    FOREIGN KEY (user_id) REFERENCES users(user_id)
);

CREATE TABLE IF NOT EXISTS answers (
    answer_id INT AUTO_INCREMENT PRIMARY KEY,
    form_id INT NOT NULL,
    users_age INT NOT NULL,
    content VARCHAR(100)  NOT NULL,
    emotion_type ENUM('Joy', 'Trust', 'Fear', 'Sadness', 'Anger', 'Anticipation', 'Surprise',
'Disgust') NOT NULL,
    created_at DATETIME NOT NULL DEFAULT CURRENT_TIMESTAMP,
    gender ENUM('male', 'female', 'other') NOT NULL,
    education_level ENUM('middle_school', 'high_school', 'college_undergraduate', 'faculty',
'masters_degree', 'phd', 'other') NOT NULL,
    experience ENUM('daily', 'weekly', 'monthly', 'yearly', 'first_time', 'never') NOT NULL,
    FOREIGN KEY (form_id) REFERENCES forms(form_id) ON DELETE CASCADE
);
```