NTNU
Kunnskap for en bedre verden

Department of Computer Science

TDT4237 Software Security and Data Privacy

Exercise 1, 2022: Information Gathering

Group 21

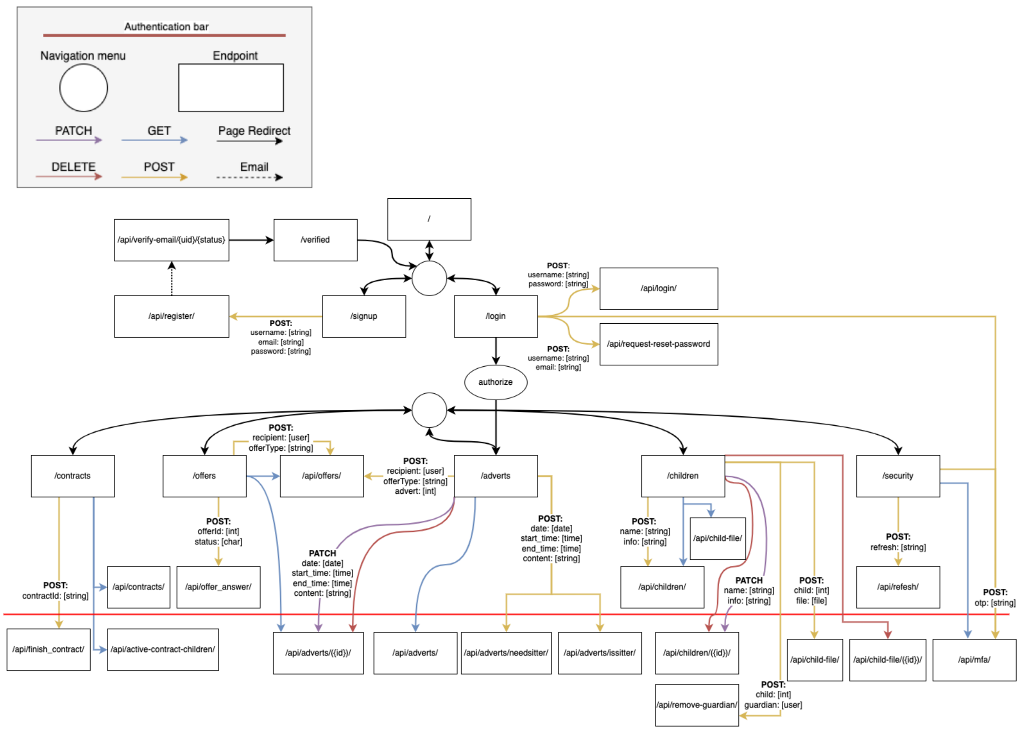
Author(s):
Patrick Øivind Helvik Legendre
Gunnar Nystad
Dag Kirstihagen

Table of Contents

1 Introduction

The objective of this report is to perform information gathering of TrustedSitters website. We will first look at the site's layout and draw up a page map using OWASP ZAP. We will then take a deeper look at the services Trustedsitters employs. This includes the website's web server, frontend technology stack, backend technology, and choice of database.

2 Page map



Authentication bar
Navigation menu
Endpoint
PATCH
GET
Page Redirect
DELETE
POST
Email
/
/api/verify-email/{uid}/{status}
/verified
/api/register/
POST: username: [string] email: [string] password: [string]
/signup
/login
POST: username: [string] password: [string]
/api/login/
POST: username: [string] email: [string]
/api/request-reset-password
authorize
/contracts
/offers
POST: recipient: [user] offerType: [string]
/api/offers/
POST: recipient: [user] offerType: [string] advert: [int]
/adverts
/children
/security
POST: offerId: [int] status: [char]
PATCH date: [date] start_time: [time] end_time: [time] content: [string]
POST: date: [date] start_time: [time] end_time: [time] content: [string]
POST: name: [string] info: [string]
/api/child-file/
POST: refresh: [string]
POST: contractId: [string]
/api/contracts/
/api/offer_answer/
/api/children/
PATCH name: [string] info: [string]
POST: child: [int] file: [file]
/api/refesh/
POST: otp: [string]
/api/finish_contract/
/api/active-contract-children/
/api/adverts/({id})/
/api/adverts/
/api/adverts/needsitter/
/api/adverts/issitter/
/api/children/({id})/
/api/child-file/
/api/child-file/({id})/
/api/mfa/
/api/remove-guardian/
POST: child: [int] guardian: [user]

Figure 1: Pagemap

2.1 Stages listed

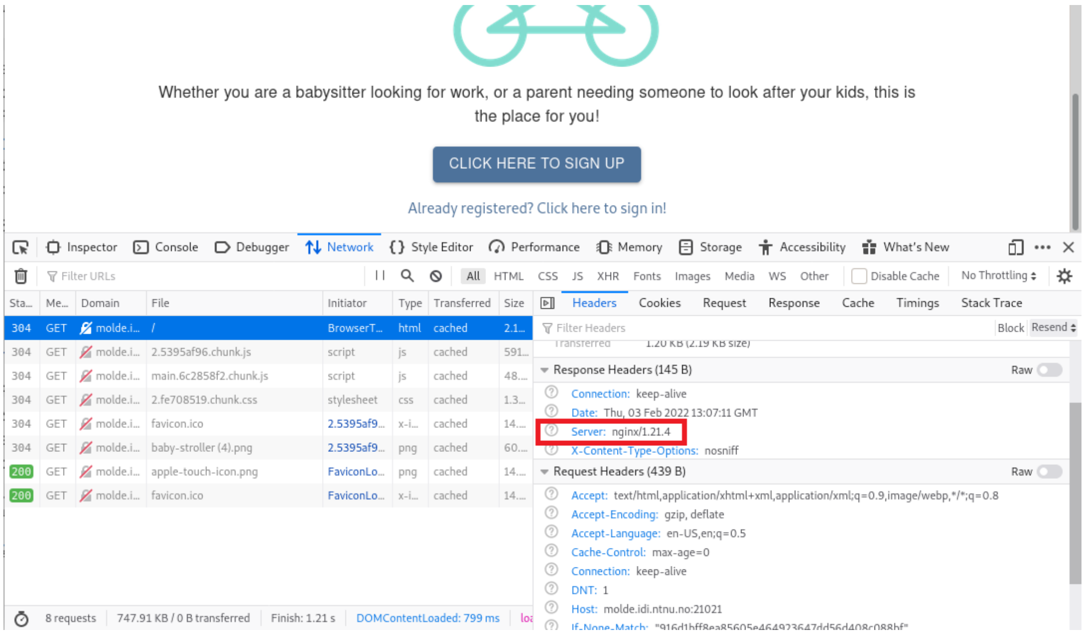
The website's page map is displayed in figure 1. We decided to follow the example provided in the exercise description in terms of labels for the page map. We maneuvered through the website using the OWASP ZAP application to find the information necessary to create the page map. To find the variable type of the POST requests, we examined the source code, request and made educated guesses as we do not have access to the database.

The user has to be authenticated to access the navigation bar, letting them navigate through the site. The authentication bar separates */api/* requests that ask for authentication and those accessed without an authentication token. Even if the user navigates the Trustedsitters website in the developers' intended way, the user would not be able to access the /contracts and /offers endpoints because it would be hidden for the user. The only way to reveal that menu would be to log in and let the server redirect you to /adverts. But anyone can access other endpoints without being logged in by directly typing in the URL. We used Postman and the browser to test each URL endpoint to find the endpoint that asked for authentication and others that did not. Every endpoint that returned an error for not being authenticated is placed under the authentication bar. We did not include /api/contracts that return a Django error because it does not explicitly return an authentication error. If the user has enabled two-factor identification for their account, a POST request is sent to /api/mfa after the username and password are entered.

3 Services

3.1 Webserver

To find out what webserver Trustedsitters uses, we can inspect the browser requests. We open the browser inspector tool and go over to the network tab that shows all requests sent to molde.idi.ntnu.no. As you can see in figure 2, the response headers in the subcategory Server say nginx/1.21.4. The webserver used on Trustedsitters is nginx.


Whether you are a babysitter looking for work, or a parent needing someone to look after your kids, this is the place for you!

Sta...	Me...	Domain	File	Initiator	Type	Transferred	Size
304	GET	molde.i...	/	BrowserT...	html	cached	2.1...
304	GET	molde.i...	2.5395af96.chunk.js	script	js	cached	591...
304	GET	molde.i...	main.6c2858f2.chunk.js	script	js	cached	48....
304	GET	molde.i...	2.fe708519.chunk.css	stylesheet	css	cached	1.3...
304	GET	molde.i...	favicon.ico	2.5395af9...	x-i...	cached	14....
304	GET	molde.i...	baby-stroller (4).png	2.5395af9...	png	cached	60....
200	GET	molde.i...	apple-touch-icon.png	FaviconLo...	png	cached	14....
200	GET	molde.i...	favicon.ico	FaviconLo...	x-i...	cached	14....

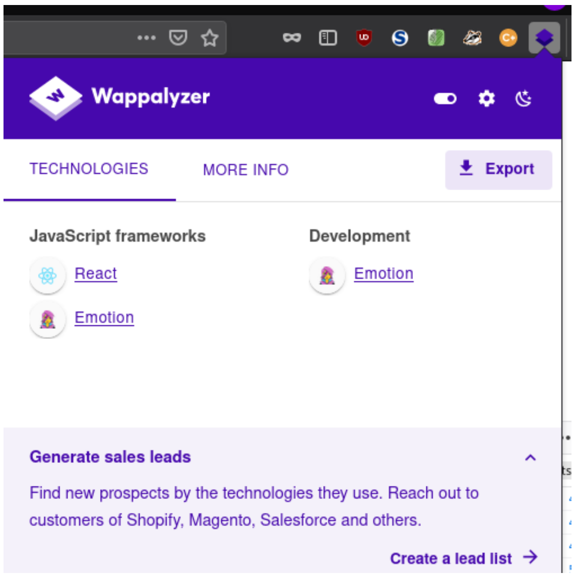
```
8 requests   747.91 KB / 0 B transferred   Finish: 1.21 s   DOMContentLoaded: 799 ms
```

```
Transferred     1.20 KB (2.19 KB size)
Response Headers (145 B)                                                   Raw
  Connection: keep-alive
  Date: Thu, 03 Feb 2022 13:07:11 GMT
  Server: nginx/1.21.4
  X-Content-Type-Options: nosniff
Request Headers (439 B)                                                    Raw
  Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
  Accept-Encoding: gzip, deflate
  Accept-Language: en-US,en;q=0.5
  Cache-Control: max-age=0
  Connection: keep-alive
  DNT: 1
  Host: molde.idi.ntnu.no:21021
  If-None-Match: "916d1bff8ea85605e464923647dd56d408c088bf"
```

Figure 2: Screenshot webserver browser http response

3.2 Frontend

We can use two options to determine which technology stack the website uses as a frontend. In the first option, the group used is a browser extension tool called Wappalyzer[1]. This tool scans the websites for different technologies it uses. By using the tool, we find out that the frontend used is React, as shown in figure 3.



JavaScript frameworks
- React
- Emotion

Development
- Emotion

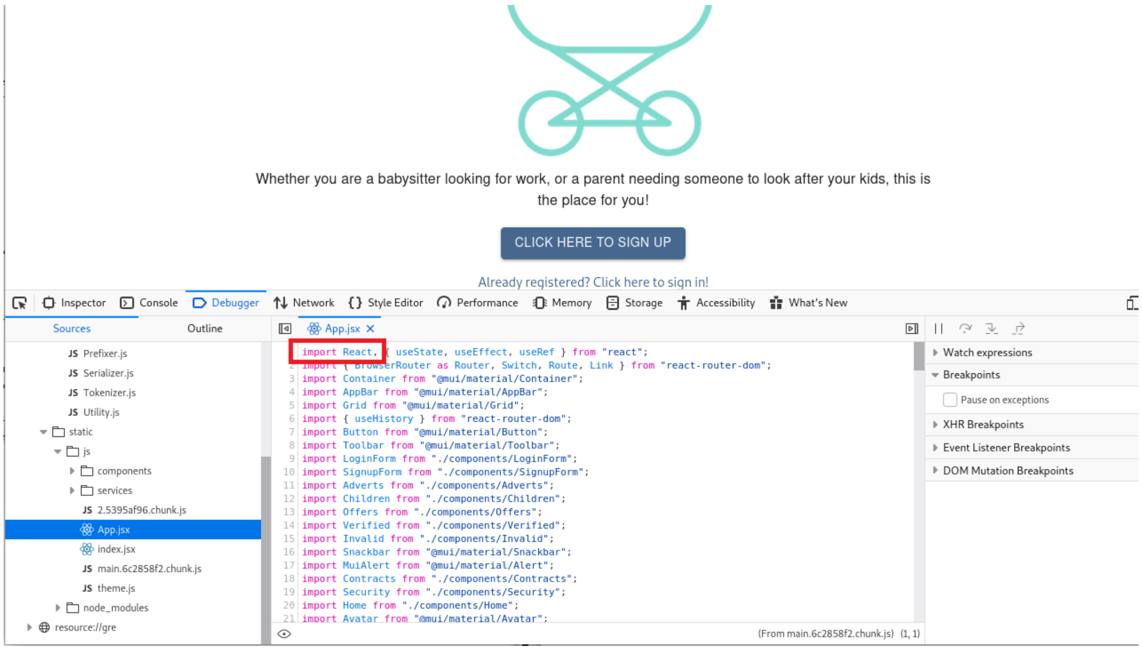
Generate sales leads

Find new prospects by the technologies they use. Reach out to customers of Shopify, Magento, Salesforce and others.

Create a lead list →

Figure 3: Screenshot wrappalyzer

The other option we can use is to go in the Debugger in the browser console. The debugger menu gives you the option to find the application's source code. We can find the source code by moving to the static folder, then under folder js, and see a file named App.jsx. We can look in the App.jsx file, and we can see in the first import statement that this file imports the React module, illustrated in figure 4.



Whether you are a babysitter looking for work, or a parent needing someone to look after your kids, this is the place for you!

```
Prefixer.js
Serializer.js
Tokenizer.js
Utility.js
static
  js
    components
    services
    2.5395af96.chunk.js
    App.jsx
    index.jsx
    main.6c2858f2.chunk.js
    theme.js
  node_modules
resource://gre
```

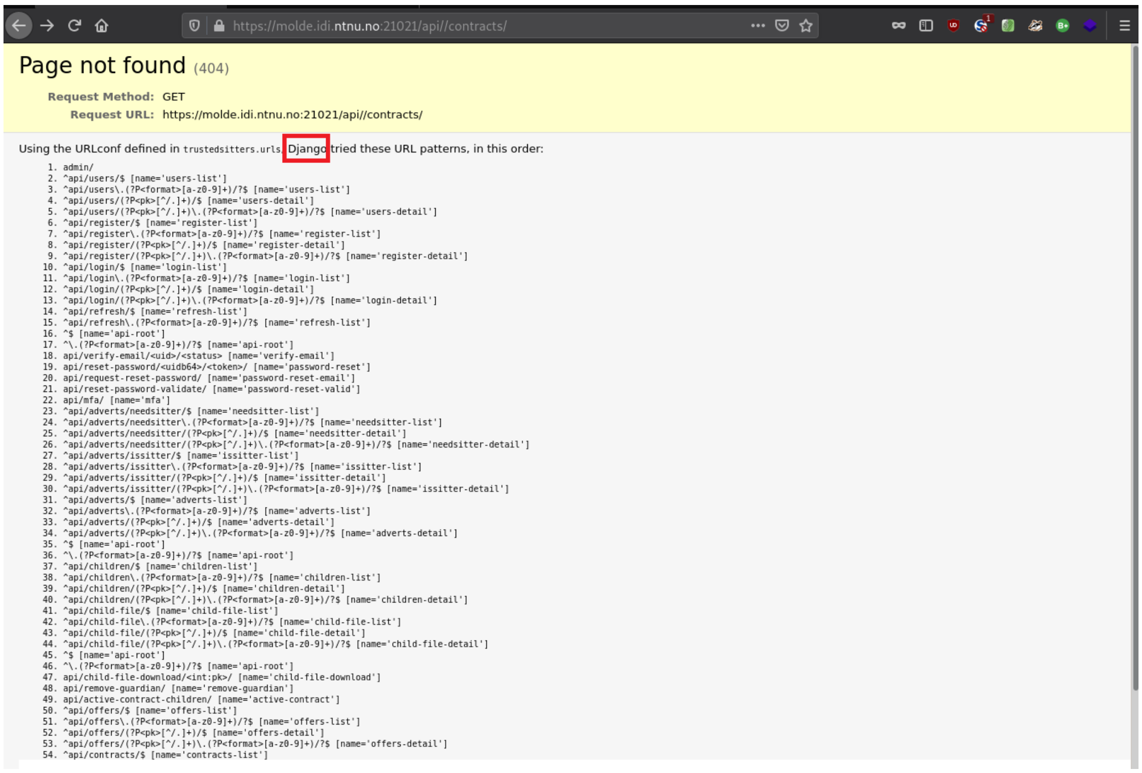
```
import React, { useState, useEffect, useRef } from "react";
import { BrowserRouter as Router, Switch, Route, Link } from "react-router-dom";
import Container from "@mui/material/Container";
import AppBar from "@mui/material/AppBar";
import Grid from "@mui/material/Grid";
import { useHistory } from "react-router-dom";
import Button from "@mui/material/Button";
import Toolbar from "@mui/material/Toolbar";
import LoginForm from "./components/LoginForm";
import SignupForm from "./components/SignupForm";
import Adverts from "./components/Adverts";
import Children from "./components/Children";
import Offers from "./components/Offers";
import Verified from "./components/Verified";
import Invalid from "./components/Invalid";
import Snackbar from "@mui/material/Snackbar";
import MuiAlert from "@mui/material/Alert";
import Contracts from "./components/Contracts";
import Security from "./components/Security";
import Home from "./components/Home";
import Avatar from "@mui/material/Avatar";
```

(From main.6c2858f2.chunk.js) (1, 1)

Figure 4: Screenshot React code

3.3 Backend

To find out the technology used for the backend for Trustedsitters, we need to check if there is an error in the code that could return very revealing errors. From the analysis of requests using the ZAP tool present on the website, we found out that the frontend uses a /api/ to send requests to the backend. If these API requests are poorly configured, the backend could return errors in its HTTP response, and we could use these HTTP response errors to gather more information about the backend. We bypass all the frontend user verification by directly sending a request to the backend. We can try to access an incorrect URL using /api/. We try using this URL: http://molde.idi.ntnu.no:21021/api//contracts/. By requesting this URL, the backend returns a 404 HTTP error but also returns more information, as shown in figure 5. In this return message, we can see all the different URLs that have been created as endpoints in the backend. We also see that this is a Django error message in this message. We also learned that this website has Django debug mode enabled with this message.



Page not found (404)

Request Method: GET
Request URL: https://molde.idi.ntnu.no:21021/api//contracts/

Using the URLconf defined in `trustedsitters.urls`, Django tried these URL patterns, in this order:

```
 1. admin/
 2. ^api/users/$ [name='users-list']
 3. ^api/users\.(?P<format>[a-z0-9]+)/?$ [name='users-list']
 4. ^api/users/(?P<pk>[^/.]+)/$ [name='users-detail']
 5. ^api/users/(?P<pk>[^/.]+)\.(?P<format>[a-z0-9]+)/?$ [name='users-detail']
 6. ^api/register/$ [name='register-list']
 7. ^api/register\.(?P<format>[a-z0-9]+)/?$ [name='register-list']
 8. ^api/register/(?P<pk>[^/.]+)/$ [name='register-detail']
 9. ^api/register/(?P<pk>[^/.]+)\.(?P<format>[a-z0-9]+)/?$ [name='register-detail']
10. ^api/login/$ [name='login-list']
11. ^api/login\.(?P<format>[a-z0-9]+)/?$ [name='login-list']
12. ^api/login/(?P<pk>[^/.]+)/$ [name='login-detail']
13. ^api/login/(?P<pk>[^/.]+)\.(?P<format>[a-z0-9]+)/?$ [name='login-detail']
14. ^api/refresh/$ [name='refresh-list']
15. ^api/refresh\.(?P<format>[a-z0-9]+)/?$ [name='refresh-list']
16. ^$ [name='api-root']
17. ^\.(?P<format>[a-z0-9]+)/?$ [name='api-root']
18. api/verify-email/<uid>/<status> [name='verify-email']
19. api/reset-password/<uidb64>/<token>/ [name='password-reset']
20. api/request-reset-password/ [name='password-reset-email']
21. api/reset-password-validate/ [name='password-reset-valid']
22. api/mfa/ [name='mfa']
23. ^api/adverts/needsitter/$ [name='needsitter-list']
24. ^api/adverts/needsitter\.(?P<format>[a-z0-9]+)/?$ [name='needsitter-list']
25. ^api/adverts/needsitter/(?P<pk>[^/.]+)/$ [name='needsitter-detail']
26. ^api/adverts/needsitter/(?P<pk>[^/.]+)\.(?P<format>[a-z0-9]+)/?$ [name='needsitter-detail']
27. ^api/adverts/issitter/$ [name='issitter-list']
28. ^api/adverts/issitter\.(?P<format>[a-z0-9]+)/?$ [name='issitter-list']
29. ^api/adverts/issitter/(?P<pk>[^/.]+)/$ [name='issitter-detail']
30. ^api/adverts/issitter/(?P<pk>[^/.]+)\.(?P<format>[a-z0-9]+)/?$ [name='issitter-detail']
31. ^api/adverts/$ [name='adverts-list']
32. ^api/adverts\.(?P<format>[a-z0-9]+)/?$ [name='adverts-list']
33. ^api/adverts/(?P<pk>[^/.]+)/$ [name='adverts-detail']
34. ^api/adverts/(?P<pk>[^/.]+)\.(?P<format>[a-z0-9]+)/?$ [name='adverts-detail']
35. ^$ [name='api-root']
36. ^\.(?P<format>[a-z0-9]+)/?$ [name='api-root']
37. ^api/children/$ [name='children-list']
38. ^api/children\.(?P<format>[a-z0-9]+)/?$ [name='children-list']
39. ^api/children/(?P<pk>[^/.]+)/$ [name='children-detail']
40. ^api/children/(?P<pk>[^/.]+)\.(?P<format>[a-z0-9]+)/?$ [name='children-detail']
41. ^api/child-file/$ [name='child-file-list']
42. ^api/child-file\.(?P<format>[a-z0-9]+)/?$ [name='child-file-list']
43. ^api/child-file/(?P<pk>[^/.]+)/$ [name='child-file-detail']
44. ^api/child-file/(?P<pk>[^/.]+)\.(?P<format>[a-z0-9]+)/?$ [name='child-file-detail']
45. ^$ [name='api-root']
46. ^\.(?P<format>[a-z0-9]+)/?$ [name='api-root']
47. api/child-file-download/<int:pk>/ [name='child-file-download']
48. api/remove-guardian/ [name='remove-guardian']
49. api/active-contract-children/ [name='active-contract']
50. ^api/offers/$ [name='offers-list']
51. ^api/offers\.(?P<format>[a-z0-9]+)/?$ [name='offers-list']
52. ^api/offers/(?P<pk>[^/.]+)/$ [name='offers-detail']
53. ^api/offers/(?P<pk>[^/.]+)\.(?P<format>[a-z0-9]+)/?$ [name='offers-detail']
54. ^api/contracts/$ [name='contracts-list']
```

Figure 5: Screenshot Django error 404 request

We can also access the admin panel to be sure and confirm that the backend is using Django. See figure 6.

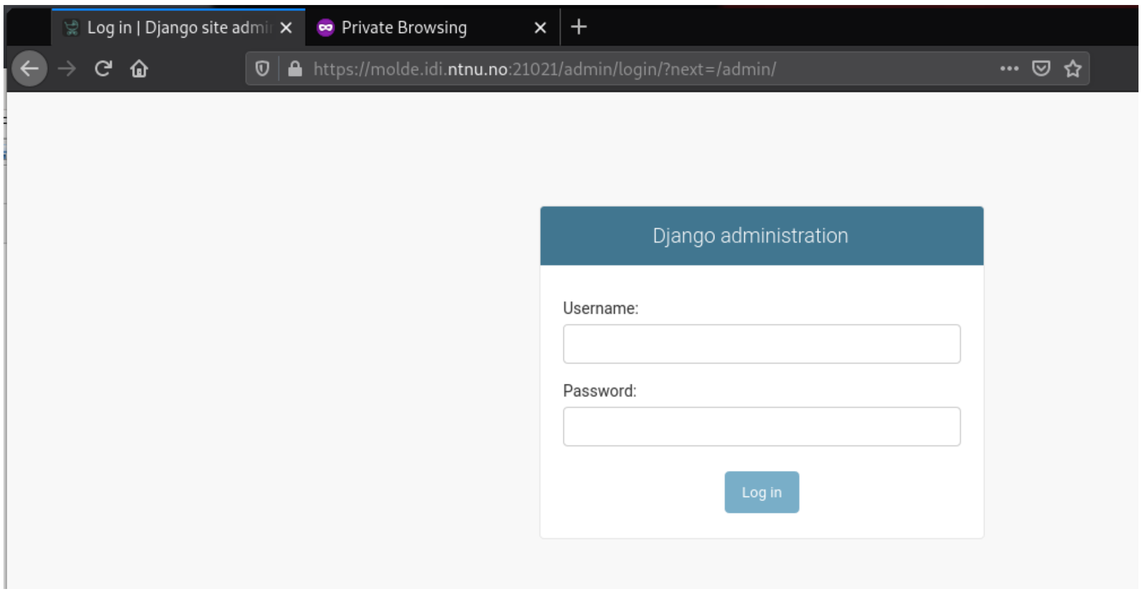
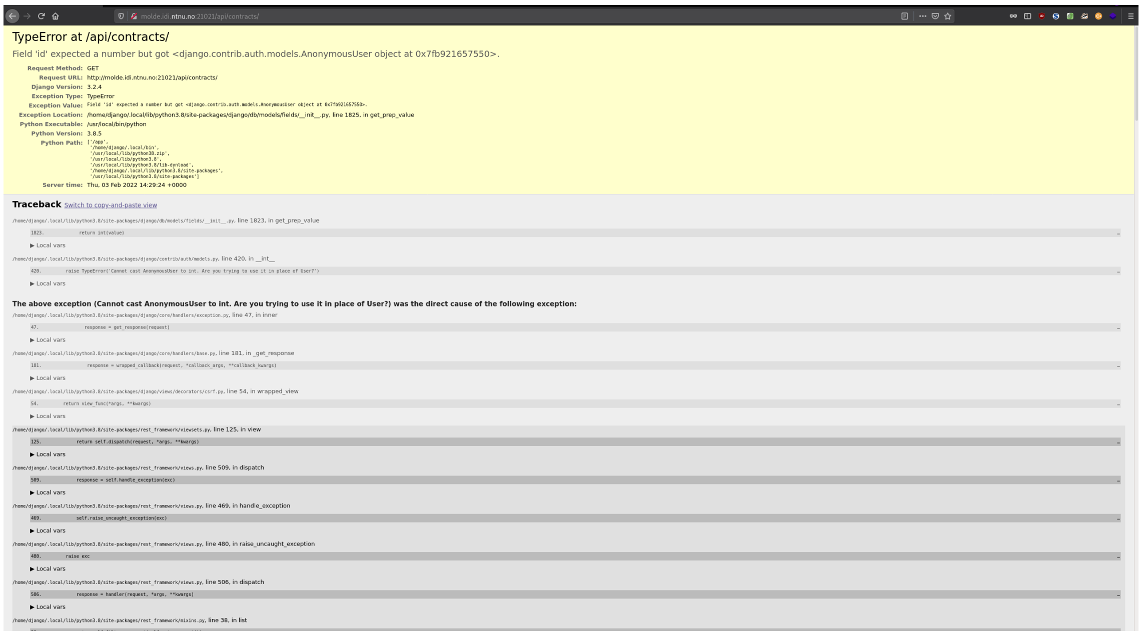

Log in | Django site admi
Private Browsing
https://molde.idi.ntnu.no:21021/admin/login/?next=/admin/
Django administration
Username:
Password:
Log in

Figure 6: Screenshot Django admin panel

3.4 Database

To find out what is used as the database is a little bit trickier. From figure 5, we got access to all the /api/ endpoints present in the server. To find out what database Django uses, we need to get more information from other more revealing error messages. We know that Django debug mode is enabled, and we got a list of all the API requests available in the backend. We are looking for an internal server error with HTTP code 500. After multiple attempts, we found that /api/contracts return us an HTTP 500 error. See figure 7. It says that Django expects an id along with the API request, but instead, Django received a django.contrib.auth.models.AnonymousUser object.

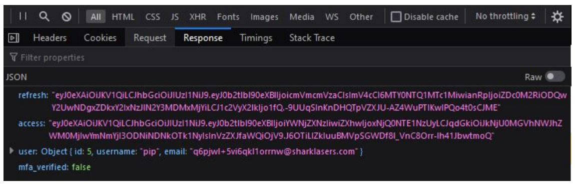


```
molde.idi.ntnu.no:21021/api/contracts/

TypeError at /api/contracts/
Field 'id' expected a number but got <django.contrib.auth.models.AnonymousUser object at 0x7fb921657550>.

      Request Method: GET
         Request URL: http://molde.idi.ntnu.no:21021/api/contracts/
      Django Version: 3.2.4
      Exception Type: TypeError
     Exception Value: Field 'id' expected a number but got <django.contrib.auth.models.AnonymousUser object at 0x7fb921657550>.
  Exception Location: /home/django/.local/lib/python3.8/site-packages/django/db/models/fields/__init__.py, line 1825, in get_prep_value
   Python Executable: /usr/local/bin/python
      Python Version: 3.8.5
         Python Path: ['/app',
                       '/home/django/.local/bin',
                       '/usr/local/lib/python38.zip',
                       '/usr/local/lib/python3.8',
                       '/usr/local/lib/python3.8/lib-dynload',
                       '/home/django/.local/lib/python3.8/site-packages',
                       '/usr/local/lib/python3.8/site-packages']
         Server time: Thu, 03 Feb 2022 14:29:24 +0000

Traceback Switch to copy-and-paste view

/home/django/.local/lib/python3.8/site-packages/django/db/models/fields/__init__.py, line 1823, in get_prep_value
    1823.             return int(value)
    ▶ Local vars
/home/django/.local/lib/python3.8/site-packages/django/contrib/auth/models.py, line 420, in __int__
    420.         raise TypeError('Cannot cast AnonymousUser to int. Are you trying to use it in place of User?')
    ▶ Local vars

The above exception (Cannot cast AnonymousUser to int. Are you trying to use it in place of User?) was the direct cause of the following exception:
/home/django/.local/lib/python3.8/site-packages/django/core/handlers/exception.py, line 47, in inner
    47.                 response = get_response(request)
    ▶ Local vars
/home/django/.local/lib/python3.8/site-packages/django/core/handlers/base.py, line 181, in _get_response
    181.                 response = wrapped_callback(request, *callback_args, **callback_kwargs)
    ▶ Local vars
/home/django/.local/lib/python3.8/site-packages/django/views/decorators/csrf.py, line 54, in wrapped_view
    54.         return view_func(*args, **kwargs)
    ▶ Local vars
/home/django/.local/lib/python3.8/site-packages/rest_framework/viewsets.py, line 125, in view
    125.             return self.dispatch(request, *args, **kwargs)
    ▶ Local vars
/home/django/.local/lib/python3.8/site-packages/rest_framework/views.py, line 509, in dispatch
    509.             response = self.handle_exception(exc)
    ▶ Local vars
/home/django/.local/lib/python3.8/site-packages/rest_framework/views.py, line 469, in handle_exception
    469.             self.raise_uncaught_exception(exc)
    ▶ Local vars
/home/django/.local/lib/python3.8/site-packages/rest_framework/views.py, line 480, in raise_uncaught_exception
    480.         raise exc
    ▶ Local vars
/home/django/.local/lib/python3.8/site-packages/rest_framework/views.py, line 506, in dispatch
    506.             response = handler(request, *args, **kwargs)
    ▶ Local vars
/home/django/.local/lib/python3.8/site-packages/rest_framework/mixins.py, line 38, in list
```

Figure 7: Screenshot Django http 500 internal server error message

This error is caused by the user not being authenticated on the website. By looking through the code, we see that the user receives a random access token used as authentication to access the

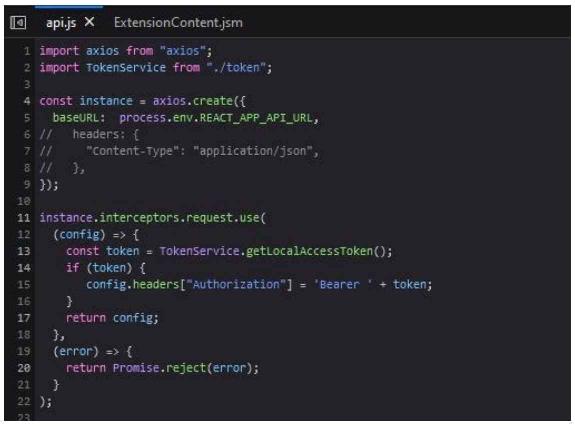
/api/ request of the application, as shown in figure 8. The token is named access in this response message.



```
refresh: "eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJ0b2tlbl90eXBlIjoicmVmcmVzaCIsImV4cCI6MTY0NTQ1MTc1MiwianRpIjoiZDc0M2RiODQw
          Y2UwNDgxZDkxY2IxNzJlN2Y3MDMxMjYiLCJ1c2VyX2lkIjo1fQ.-9UUqSlnKnDHQTpVZXJU-AZ4WuPTIKwIPQo4t0sCJME"
access: "eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJ0b2tlbl90eXBlIjoiYWNjZXNzIiwiZXhwIjoxNjQ0NTE1NzUyLCJqdGkiOiJkNjU0MGVhNWJhZ
         WM0MjIwYmNmYjI3ODNiNDNkOTk1NyIsInVzZXJfaWQiOjV9.J6OTiLIZkluuBMVpSGWDf8I_VnC8Orr-Ih41JbwtmoQ"
▶ user: Object { id: 5, username: "pip", email: "q6pjwl+5vi6qkl1orrnw@sharklasers.com" }
mfa_verified: false
```

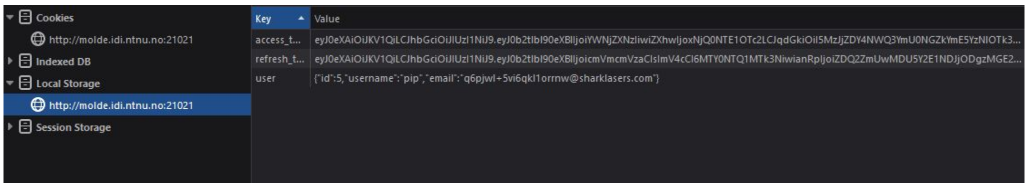
Figure 8: Screenshot response request access token

This token is included in the request when it is send from the frontend by invoking the method TokenService.getLocalAccessToken(); See figure 9. This method fetches the token stored in the local storage of the browser, illustrated in figure 10.



```
import axios from "axios";
import TokenService from "./token";

const instance = axios.create({
  baseURL:  process.env.REACT_APP_API_URL,
//   headers: {
//     "Content-Type": "application/json",
//   },
});

instance.interceptors.request.use(
  (config) => {
    const token = TokenService.getLocalAccessToken();
    if (token) {
      config.headers["Authorization"] = 'Bearer ' + token;
    }
    return config;
  },
  (error) => {
    return Promise.reject(error);
  }
);
```

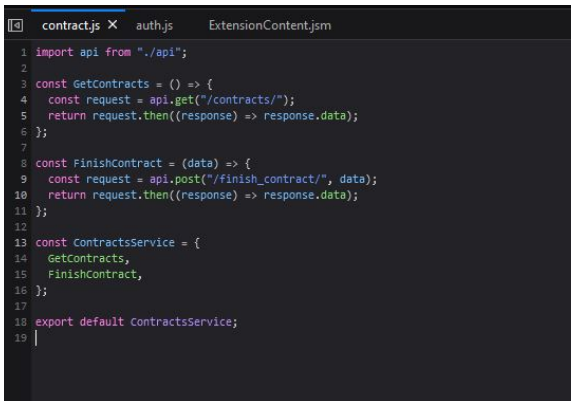
Figure 9: Screenshot api.js frontend code


Cookies
http://molde.idi.ntnu.no:21021
Indexed DB
Local Storage
http://molde.idi.ntnu.no:21021
Session Storage

Key	Value
access_t...	eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJ0b2tlbl90eXBlIjoiYWNjZXNzIiwiZXhwIjoxNjQ0NTE1OTc2LCJqdGkiOiI5MzJjZDY4NWQ3YmU0NGZkYmE5YzNlOTk3...
refresh_t...	eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJ0b2tlbl90eXBlIjoicmVmcmVzaCIsImV4cCI6MTY0NTQ1MTk3NiwianRpIjoiZDQ2ZmUwMDU5Y2E1NDJjODgzMGE2...
user	{"id":5,"username":"pip","email":"q6pjwl+5vi6qkl1orrnw@sharklasers.com"}

Figure 10: Screenshot browser local storage

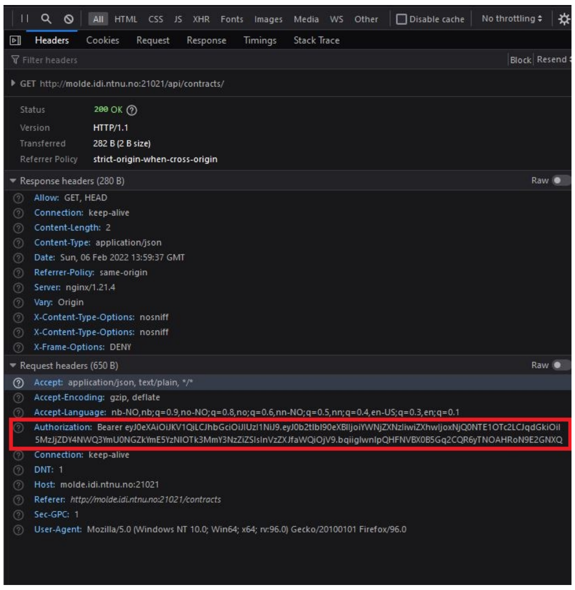
When the frontend source code is calling the backend uses the api interceptor to fetch contracts. It calls the method api.get("/contracts/") which request the api/contracts/ endpoint. See figure 11.



```
import api from "./api";

const GetContracts = () => {
  const request = api.get("/contracts/");
  return request.then((response) => response.data);
};

const FinishContract = (data) => {
  const request = api.post("/finish_contract/", data);
  return request.then((response) => response.data);
};

const ContractsService = {
  GetContracts,
  FinishContract,
};

export default ContractsService;
```

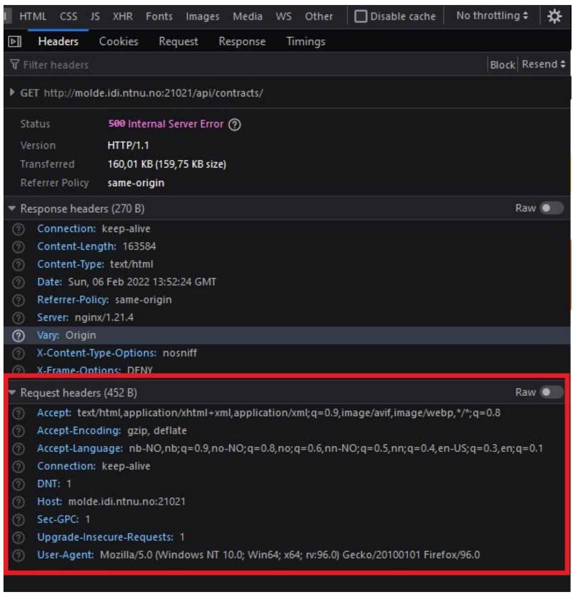
Figure 11: Screenshot contract.js frontend code

The request sent to the backend is shown in figure 12. But if the browser does not have an access token stored in the local storage and we still try to access the request, then we receive the internal server HTTP error. See figure 13. This is because the backend was expecting the access token and could fetch the user id to retrieve all the contracts associated with that user.



```
GET http://molde.idi.ntnu.no:21021/api/contracts/

Status           200 OK
Version          HTTP/1.1
Transferred      282 B (2 B size)
Referrer Policy  strict-origin-when-cross-origin

Response headers (280 B)
Allow: GET, HEAD
Connection: keep-alive
Content-Length: 2
Content-Type: application/json
Date: Sun, 06 Feb 2022 13:59:37 GMT
Referrer-Policy: same-origin
Server: nginx/1.21.4
Vary: Origin
X-Content-Type-Options: nosniff
X-Content-Type-Options: nosniff
X-Frame-Options: DENY

Request headers (650 B)
Accept: application/json, text/plain, */*
Accept-Encoding: gzip, deflate
Accept-Language: nb-NO,nb;q=0.9,no-NO;q=0.8,no;q=0.6,nn-NO;q=0.5,nn;q=0.4,en-US;q=0.3,en;q=0.1
Authorization: Bearer eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJ0b2tlbl90eXBlIjoiYWNjZXNzIiwiZXhwIjoxNjQ0NTE1OTc2LCJqdGkiOiI
5MzJjZDY4NWQ3YmU0NGZkYmE5YzNlOTk3MmY3NzZiZSIsInVzZXJfaWQiOjV9.bqiiglwnlpQHFNVBX0B5Gq2CQR6yTNOAHRoN9E2GNXQ
Connection: keep-alive
DNT: 1
Host: molde.idi.ntnu.no:21021
Referer: http://molde.idi.ntnu.no:21021/contracts
Sec-GPC: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:96.0) Gecko/20100101 Firefox/96.0
```

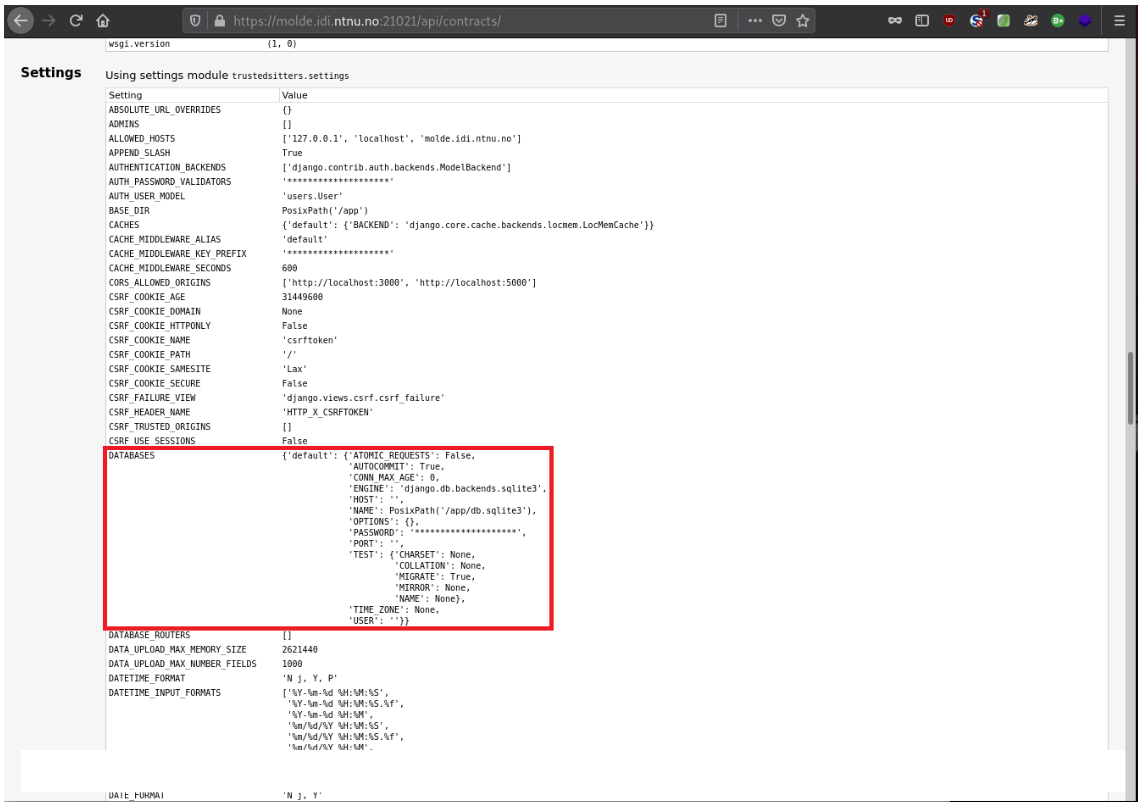
Figure 12: Screenshot authorization request /api/contact



```
GET http://molde.idi.ntnu.no:21021/api/contracts/

Status           500 Internal Server Error
Version          HTTP/1.1
Transferred      160,01 KB (159,75 KB size)
Referrer Policy  same-origin

Response headers (270 B)
Connection: keep-alive
Content-Length: 163584
Content-Type: text/html
Date: Sun, 06 Feb 2022 13:52:24 GMT
Referrer-Policy: same-origin
Server: nginx/1.21.4
Vary: Origin
X-Content-Type-Options: nosniff
X-Frame-Options: DENY

Request headers (452 B)
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
Accept-Encoding: gzip, deflate
Accept-Language: nb-NO,nb;q=0.9,no-NO;q=0.8,no;q=0.6,nn-NO;q=0.5,nn;q=0.4,en-US;q=0.3,en;q=0.1
Connection: keep-alive
DNT: 1
Host: molde.idi.ntnu.no:21021
Sec-GPC: 1
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:96.0) Gecko/20100101 Firefox/96.0
```

Figure 13: Screenshot 500 internal server error request /api/contact

This is why we receive the Django HTTP 500 error. By scrolling down the error page, we can find other information about the backend and the code used in the backend. But if we scroll down to the database section, we can see which database it uses to store data. In the "DATABASES" section, we see that the database referenced is sqlite3. See figure 14. So the database used is SQLite.



https://molde.idi.ntnu.no:21021/api/contracts/

```
wsgi.version                 (1, 0)
```

Settings

Using settings module `trustedsitters.settings`

```
Setting                      Value
ABSOLUTE_URL_OVERRIDES       {}
ADMINS                       []
ALLOWED_HOSTS                ['127.0.0.1', 'localhost', 'molde.idi.ntnu.no']
APPEND_SLASH                 True
AUTHENTICATION_BACKENDS      ['django.contrib.auth.backends.ModelBackend']
AUTH_PASSWORD_VALIDATORS     '********************'
AUTH_USER_MODEL              'users.User'
BASE_DIR                     PosixPath('/app')
CACHES                       {'default': {'BACKEND': 'django.core.cache.backends.locmem.LocMemCache'}}
CACHE_MIDDLEWARE_ALIAS       'default'
CACHE_MIDDLEWARE_KEY_PREFIX  '********************'
CACHE_MIDDLEWARE_SECONDS     600
CORS_ALLOWED_ORIGINS         ['http://localhost:3000', 'http://localhost:5000']
CSRF_COOKIE_AGE              31449600
CSRF_COOKIE_DOMAIN           None
CSRF_COOKIE_HTTPONLY         False
CSRF_COOKIE_NAME             'csrftoken'
CSRF_COOKIE_PATH             '/'
CSRF_COOKIE_SAMESITE         'Lax'
CSRF_COOKIE_SECURE           False
CSRF_FAILURE_VIEW            'django.views.csrf.csrf_failure'
CSRF_HEADER_NAME             'HTTP_X_CSRFTOKEN'
CSRF_TRUSTED_ORIGINS         []
CSRF_USE_SESSIONS            False
DATABASES                    {'default': {'ATOMIC_REQUESTS': False,
                                          'AUTOCOMMIT': True,
                                          'CONN_MAX_AGE': 0,
                                          'ENGINE': 'django.db.backends.sqlite3',
                                          'HOST': '',
                                          'NAME': PosixPath('/app/db.sqlite3'),
                                          'OPTIONS': {},
                                          'PASSWORD': '********************',
                                          'PORT': '',
                                          'TEST': {'CHARSET': None,
                                                   'COLLATION': None,
                                                   'MIGRATE': True,
                                                   'MIRROR': None,
                                                   'NAME': None},
                                          'TIME_ZONE': None,
                                          'USER': ''}}
DATABASE_ROUTERS             []
DATA_UPLOAD_MAX_MEMORY_SIZE  2621440
DATA_UPLOAD_MAX_NUMBER_FIELDS 1000
DATETIME_FORMAT              'N j, Y, P'
DATETIME_INPUT_FORMATS       ['%Y-%m-%d %H:%M:%S',
                              '%Y-%m-%d %H:%M:%S.%f',
                              '%Y-%m-%d %H:%M',
                              '%m/%d/%Y %H:%M:%S',
                              '%m/%d/%Y %H:%M:%S.%f',
                              '%m/%d/%Y %H:%M',
DATE_FORMAT                  'N j, Y'
```

Figure 14: Screenshot over Django error message database list

4 Conclusion

We managed to map out the Trustedsitters website by finding most of the relevant endpoints. We also managed to find out the webserver, the frontend, the backend, and the database used for this website. The group found out that Django was misconfigured and exposed sensitive information about the web application. To prevent the web application from leaking data and provide better security overall. The web application should be hosted on an HTTPS domain with a certificate where someone malicious could not intercept the traffic. Each API request should be protected by checking the authentication token of an authenticated user. The Trustedsitters should also disable debugging mode in Django because this leaks information about the application we do not want to expose to users. Looking at the pagemap we can conclude that the authentication bar should be higher up and Trustedsitters should protect with authentication more of the api request to provide better security for both the webserver and the users.

References

[1] wappalyzer. *wappalyzer*. 2022. URL: `https://www.wappalyzer.com/` (visited on 02/06/2022).