**Bezpieczeństwo aplikacji mobilnych**

**PROJEKT**

***Menadżer haseł***

Jakub Jach  
Patryk Jaworski  
1ID24A

1. **Wykorzystane technologie**

W ramach projektu wykorzystano następujące technologie:

Część backend – napisana w języku Python w stylu API REST:

* django – framework do tworzenia aplikacji internetowych i webowych w oparciu o protokół HTTP
* django-otp – biblioteka umożliwiająca użycie mechanizmu OTP w ramach aplikacji django
* django-cryptography – pozwala na szyfrowanie danych zapisywanych w modelu ORM dla modeli django
* django-rest-framework – framework umożliwiający tworzenie interfejsów API REST z użyciem django – udostępnia serializery pozwalające na zwracanie obiektów i wybranych pól w postaci JSON
* base64 – umożliwia zakodowanie danych binarnych do odpowiedniej postaci – wykorzystywane do zapisywania tajnego klucza urządzenia do bazy
* environ – biblioteka pozwalająca na definiowanie zmiennych środowiskowych w zewnętrznych plikach co pozwala uniknąć problemów z hardkodowaniem

Właściwa aplikacja mobilna – napisana w języku JavaScript z użyciem Expo:

* react-native – framework do tworzenia aplikacji mobilnych z użyciem języka JavaScript
* react-native-async-storage – pozwala na dostęp do lokalnej pamięci z poziomu JavaScript
* react-native-clipboard – umożliwia dostęp do schowka np. skopiowanie do niego tekstu
* react-native-picker – biblioteka do stworzenia menu z opcjami do wyboru
* react-native-biometrics – pozwala na dostęp do biometrii urządzenia
* react-native-gesture-handler – obsługa gestów
* react-native-reanimated – animacje
* react-navigation/drawer – biblioteka do tworzenia nawigacji typu drawer
* react-navigation/stack – nawigacja typu stack
* axios – biblioteka do komunikacji z serwerem HTTP / HTTPS
* expo-local-authentication – biblioteka do lokalnego logowania, używana w połączeniu z biometrią
* expo-secure-store – biblioteka do obsługi mechanizmów bezpiecznego lokalnego przechowywania danych
* expo-status-bar – biblioteka do obsługi status bar w aplikacji

1. **Uruchomienie projektu – backend**

W celu uruchomienia części backend projektu konieczne jest zainstalowanie wymienionych bibliotek w systemie. Aby uniknąć konfliktów z już zainstalowanymi bibliotekami zalecane jest stworzenie nowego wirtualnego środowiska języka Python z użyciem menadżera pakietów *conda* lub narzędzia *venev*.

Poniżej przedstawiono przykład stworzenia nowego środowiska conda wraz z instalacją wymaganych bibliotek z pliku *conda*\_*requirements.txt*

BAM\_PRO\_JACH\_JAWORSKI$ conda create --name <nazwa> --file conda\_requiremets.txt

Po stworzeniu środowiska należy przejść do folderu *backend* z poziomu głównego folderu repozytorium (ewentualnie aktywować stworzone wirtualne środowisko). Następnie należy skonfigurować zmienne środowiskowe w pliku *.env* w podfolderze *backend* według schematu z pliku *.env.schema*. Zmienna *HOST* powinna być zgodna z nazwą DNS serwera albo jego adresem IP. Z kolei zmienna *SECRET\_KEY* jest tajnym kluczem kryptograficznym używanym przez django do generowania tokenów i szyfrowania danych. Do wygenerowania klucza należy użyć funkcji get\_random\_secret\_key z django.core.management.utils.

Kolejnym krokiem jest dokonanie migracji bazy danych w celu utworzenia wymaganych tabel:

../bam\_pro\_jach\_jaworski/backend$ python manage.py makemigrations

Migrations for 'mainApp':

mainApp\migrations\0002\_creditstorage.py

- Create model CreditStorage

../bam\_pro\_jach\_jaworski/backend$ python manage.py migrate

Operations to perform:

Apply all migrations: admin, auth, contenttypes, django\_otp, mainApp, sessions

Running migrations:

Applying contenttypes.0001\_initial... OK

Applying auth.0001\_initial... OK

Applying admin.0001\_initial... OK

Applying admin.0002\_logentry\_remove\_auto\_add... OK

Applying admin.0003\_logentry\_add\_action\_flag\_choices... OK

Applying contenttypes.0002\_remove\_content\_type\_name... OK

Applying auth.0002\_alter\_permission\_name\_max\_length... OK

Applying auth.0003\_alter\_user\_email\_max\_length... OK

Applying auth.0004\_alter\_user\_username\_opts... OK

Applying auth.0005\_alter\_user\_last\_login\_null... OK

Applying auth.0006\_require\_contenttypes\_0002... OK

Applying auth.0007\_alter\_validators\_add\_error\_messages... OK

Applying auth.0008\_alter\_user\_username\_max\_length... OK

Applying auth.0009\_alter\_user\_last\_name\_max\_length... OK

Applying auth.0010\_alter\_group\_name\_max\_length... OK

Applying auth.0011\_update\_proxy\_permissions... OK

Applying auth.0012\_alter\_user\_first\_name\_max\_length... OK

Applying django\_otp.0001\_initial... OK

Applying mainApp.0001\_initial... OK

Applying mainApp.0002\_creditstorage... OK

Applying sessions.0001\_initial... OK

Następnie można utworzyć użytkownika typu superuser, który będzie miał dostęp do strony administracyjnej django.

../bam\_pro\_jach\_jaworski/backend$ python manage.py createsuperuser

Username (leave blank to use 'domowy'): root

Email address: root@localhost.pl

Password:

Password (again):

Superuser created successfully.

Po utworzeniu nowego użytkownika można uruchomić serwer backend:

../bam\_pro\_jach\_jaworski/backend$ python manage.py runserver

Watching for file changes with StatReloader

Performing system checks...

System check identified no issues (0 silenced).

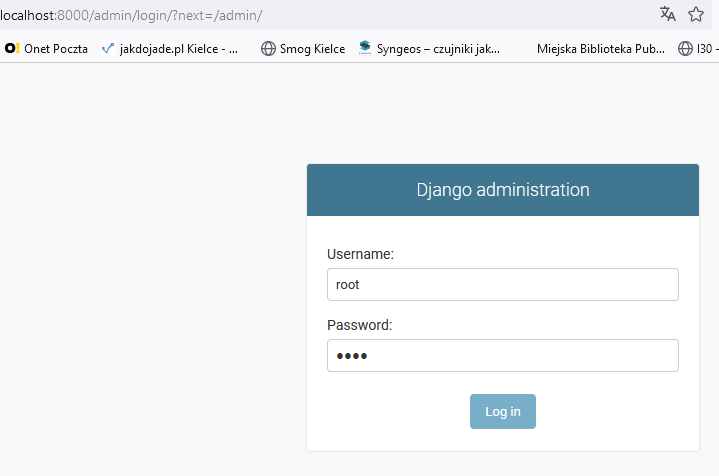
December 08, 2023 - 13:22:36

Django version 4.1, using settings 'backend.settings'

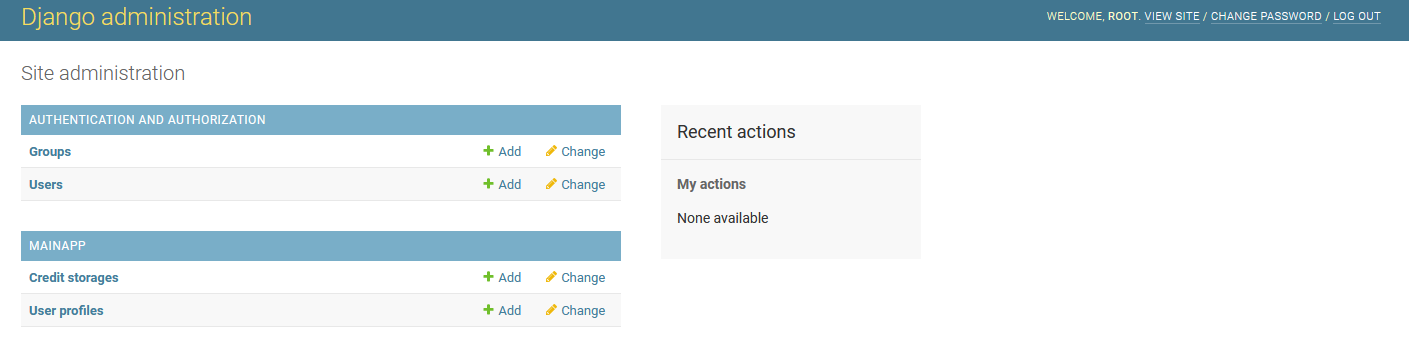
Starting development server at http://127.0.0.1:8000/

Quit the server with CTRL-BREAK.

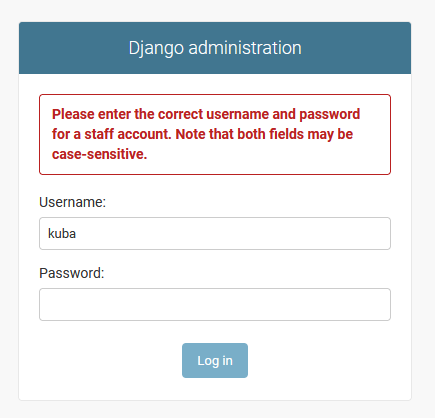
Aby uzyskać dostep do strony administracyjnej konieczne jest otwarcie strony */admin* i podanie danych do logowania:



Po zalogowaniu pokazany zostanie główny widok strony administracyjnej



Możliwe jest zarządzanie użytkownikami, podgląd zapisanych danych i ich modyfikacja, dodawanie i usuwanie. Dostęp do strony jest możliwy jedynie dla użytkowników z uprawnieniami administratora. W przeciwnym wypadku strona odmówi dostępu.



**Uruchomienie serwera z dostępem z poziomu aplikacji mobilnej**

Domyślnie serwer django uruchamia się na adresie 127.0.0.1 (localhost) co uniemożliwia dostęp z zewnątrz. Aby uruchomić projekt na innym porcie należy podać parę *adres:port* podczas uruchamiania.

../bam\_pro\_jach\_jaworski/backend$ python manage.py runserver 0.0.0.0:8000

Watching for file changes with StatReloader

Performing system checks...

System check identified no issues (0 silenced).

December 08, 2023 - 13:32:46

Django version 4.1, using settings 'backend.settings'

Starting development server at http://0.0.0.0:8000/

Quit the server with CTRL-BREAK.

Tak uruchomiony serwer umożliwia połączenie się z nim przez aplikację mobilną.

**Uruchomienie aplikacji mobilnej**

Aplikację mobilną można uruchomić za pośrednictwem Expo, które udostępnia serwer deweloperski języka JavaScript (nodejs), do którego łączymy się z użyciem aplikacji Expo zainstalowanej na telefonie.

Przed uruchomieniem aplikacji konieczne jest zainstalowanie wymaganych bibliotek

../bam\_pro\_jach\_jaworski/frontend$ npm i

added 1255 packages, and audited 1256 packages in 1m

73 packages are looking for funding

run `npm fund` for details

6 moderate severity vulnerabilities

To address issues that do not require attention, run:

npm audit fix

To address all issues (including breaking changes), run:

npm audit fix --force

Run `npm audit` for details.

Po zainstalowaniu wymaganych bibliotek należy skonfigurować zmienną środowiskową *EXPO\_PUBLIC\_API\_URL* w pliku .env (plik należy utworzyć ręcznie). Zmiennej tej należy przypisać wartość będącą nazwą komputera, na którym uruchomiony został serwer django lub jego adres IP i numer portu np. EXPO\_PUBLIC\_API\_URL=192.168.1.15:8000

Przed kolejnymi etapami konieczne jest zalogowanie się na konto Expo:

../bam\_pro\_jach\_jaworski/frontend$ npx expo login

Log in to EAS

√ Email or username ... root@localhost.com

√ Password ... \*\*\*\*\*\*\*\*\*\*\*

Po skonfigurowaniu środowiska można uruchomić serwer deweloperski aplikacji:

../bam\_pro\_jach\_jaworski/frontend$ npx expo start

Starting project at *(...)*bam\_pro\_jach\_jaworski\frontend

env: load .env

env: export EXPO\_PUBLIC\_API\_URL

Starting Metro Bundler

Some dependencies are incompatible with the installed expo version:

expo-local-authentication@13.6.0 - expected version: ~13.4.1

react-native@0.72.5 - expected version: 0.72.6

Your project may not work correctly until you install the correct versions of the packages.

Fix with: npx expo install --fix

▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄

█ ▄▄▄▄▄ █ █▄ █▄▀█ ▄▄▄▄▄ █

█ █ █ █ ▀▄ █▀▄ ▄█ █ █ █

█ █▄▄▄█ █▀██▀▀█▀▄██ █▄▄▄█ █

█▄▄▄▄▄▄▄█▄▀▄█ █▄█▄█▄▄▄▄▄▄▄█

█▄ █▀▄▄▄██▀▀▄▀█▄ ███ ▀▄▄ ▄█

█▄█▄▀▀▀▄ ▀█▀ ▄██ ▀▀ █▄ ▀██

█ █ ▀█▄ █▀▄█▄▀▄▀▄▀▄▀▀▄ ▀██

███ █▀ ▄█ ▀█▀██▄▄▄█▄▀ ▀███

█▄▄█▄▄▄▄▄▀ ▀▄▀█▄▄ ▄▄▄ ▀ ▄▄█

█ ▄▄▄▄▄ █▀▀ ▄██▀ █▄█ ▀▀█▀█

█ █ █ █▄███▄▀▄█▄▄ ▄▄▀ █

█ █▄▄▄█ █▀▀ █▀█▀▄██▄▀█▀▀ ██

█▄▄▄▄▄▄▄█▄▄█▄▄▄▄████▄▄▄▄▄▄█

› Metro waiting on exp://192.168.1.15:8081

› Scan the QR code above with Expo Go (Android) or the Camera app (iOS)

› Using Expo Go

› Press s │ switch to development build

› Press a │ open Android

› Press w │ open web

› Press j │ open debugger

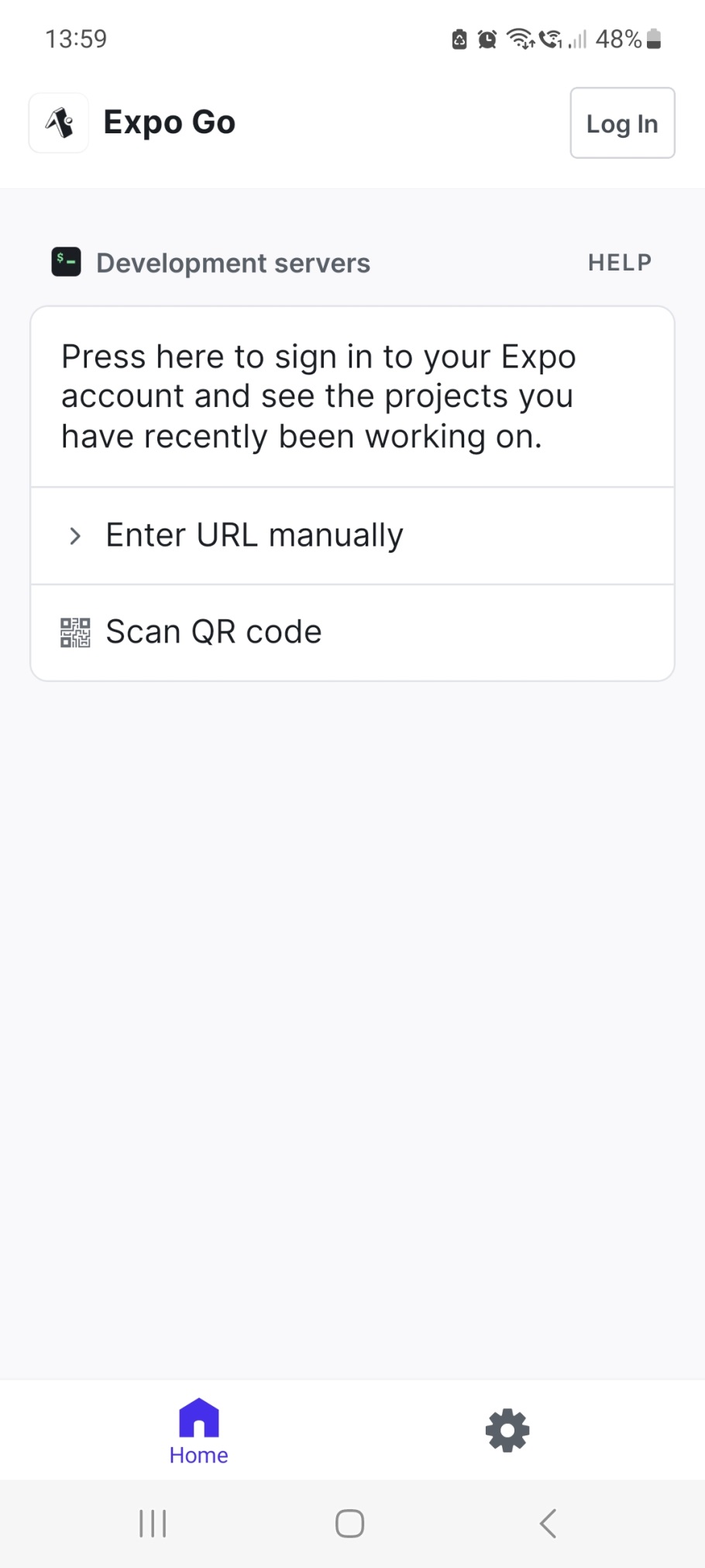
› Press r │ reload app

› Press m │ toggle menu

› Press o │ open project code in your editor

› Press ? │ show all commands

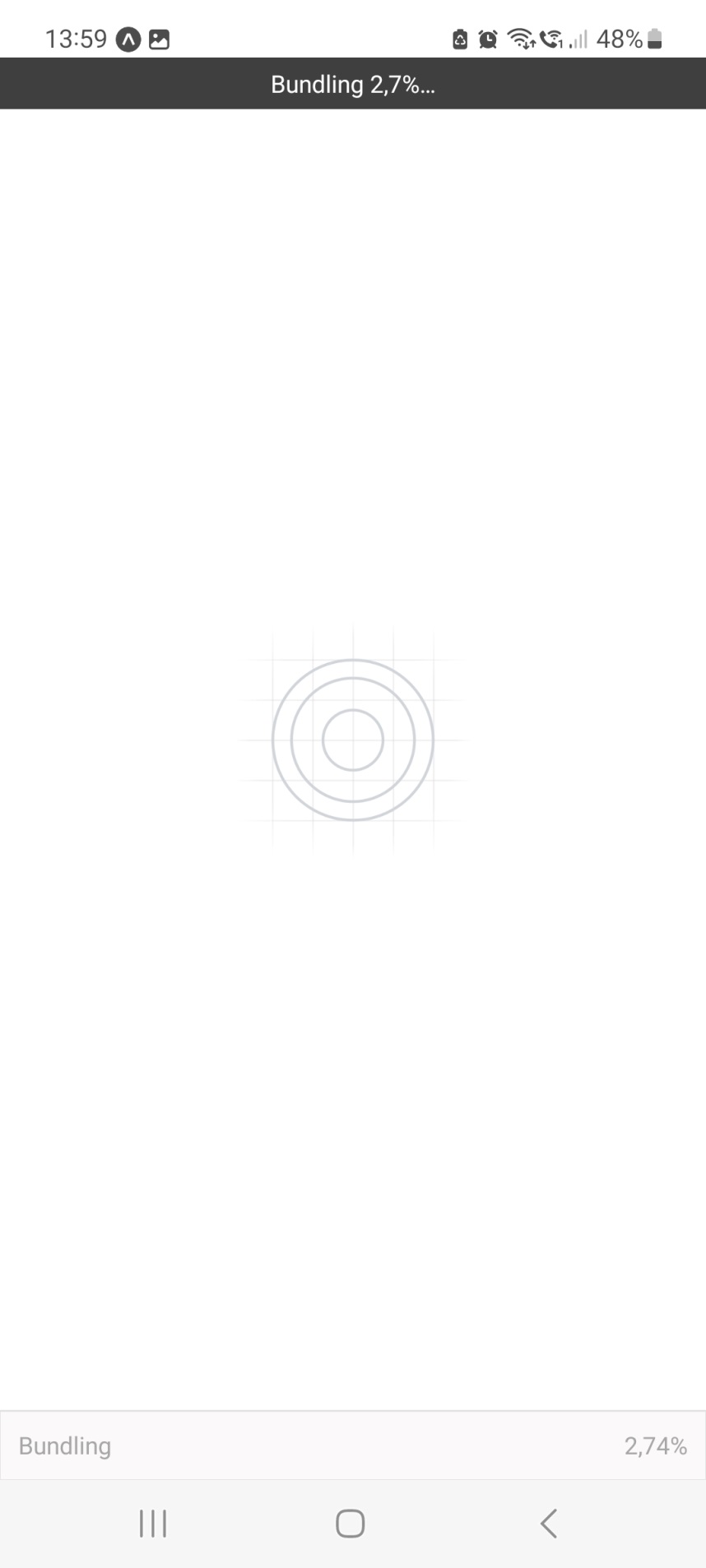
Aplikacja Expo Go otwarta na rzeczywistym urządzeniu.



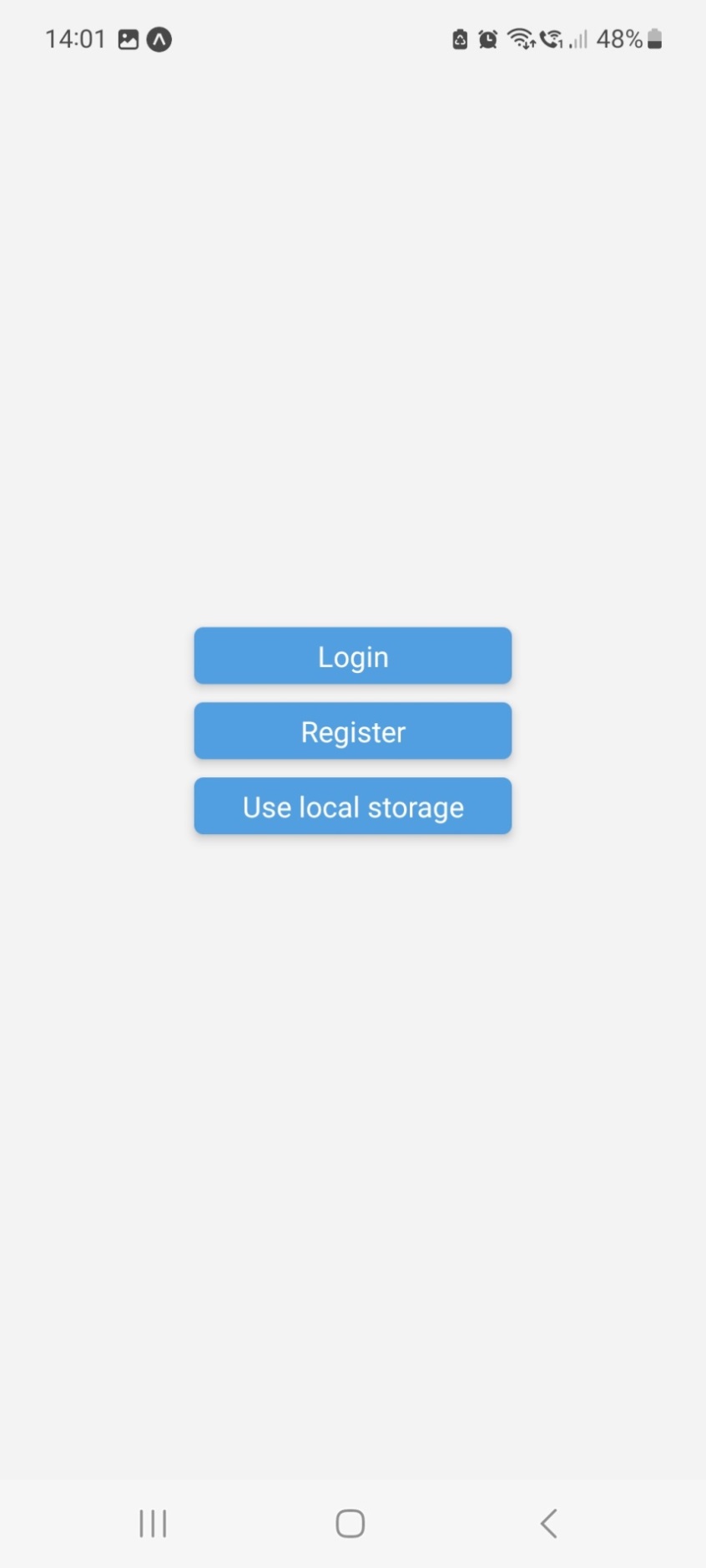
Po zeskanowaniu kodu QR rozpocznie się „bundlowanie” aplikacji:

Android node\_modules\expo/AppEntry.js ▓░░░░░░░░░░░░░░░ 8.6% (120/486)

Z kolei na ekranie telefonu również ukaże się stosowna informacja:



Po zbudowaniu aplikacji zostanie ona otwarta na telefonie:



Z kolei w konsoli pojawi się komunikat o zakończeniu budowania aplikacji:

Android Bundling complete 39161ms

**Budowanie aplikacji do dystrybucji**

Po przetestowaniu aplikacji możliwe jest zbudowanie jej do pliku *.aab* lub *.apk* w celu zainstalowania na systemie Android (na przykładzie lokalnej kompilacji do pliku *.apk*):

../frontend$: eas build --platform android --local --profile preview

Loaded "env" configuration for the "preview" profile: no environment variables specified. Learn more: https://docs.expo.dev/build-reference/variables/

✔ Using remote Android credentials (Expo server)

✔ Using Keystore from configuration: Build Credentials 2cYxBCzwAP (default)

✔ Compressed project files 12s (31.3 MB)

*(...pominięto...)*

[PREBUILD] - Config syncing

[PREBUILD] ✔ Config synced

[PREBUILD] Running "npm install" in /tmp/jakub/eas-build-local-nodejs/60fae58b-2888-4492-bb9e-8c6721f66896/build/frontend directory

[PREBUILD] added 1 package, and audited 1258 packages in 3s

[PREBUILD]

[PREBUILD] 74 packages are looking for funding

[PREBUILD] run `npm fund` for details

[PREBUILD] 7 moderate severity vulnerabilities

[PREBUILD]

[PREBUILD] To address issues that do not require attention, run:

[PREBUILD] npm audit fix

[PREBUILD]

[PREBUILD] To address all issues (including breaking changes), run:

[PREBUILD] npm audit fix --force

[PREBUILD]

[PREBUILD] Run `npm audit` for details.

[PREPARE\_CREDENTIALS] Writing secrets to the project's directory

[PREPARE\_CREDENTIALS] Injecting signing config into build.gradle

[RUN\_GRADLEW] Running 'gradlew :app:assembleRelease' in /tmp/jakub/eas-build-local-nodejs/60fae58b-2888-4492-bb9e-8c6721f66896/build/frontend/android

[RUN\_GRADLEW] Starting a Gradle Daemon, 1 incompatible and 1 stopped Daemons could not be reused, use --status for details

*(...pominięto...)*

[RUN\_GRADLEW] Using expo modules

[RUN\_GRADLEW] - expo-application (5.3.1)

[RUN\_GRADLEW] - expo-clipboard (4.3.1)

[RUN\_GRADLEW] - expo-constants (14.4.2)

[RUN\_GRADLEW] - expo-file-system (15.4.4)

[RUN\_GRADLEW] - expo-font (11.4.0)

[RUN\_GRADLEW] - expo-keep-awake (12.3.0)

[RUN\_GRADLEW] - expo-local-authentication (13.6.0)

[RUN\_GRADLEW] - expo-modules-core (1.5.11)

[RUN\_GRADLEW] - expo-modules-core$android-annotation (1.5.11)

[RUN\_GRADLEW] - expo-modules-core$android-annotation-processor (1.5.11)

[RUN\_GRADLEW] - expo-secure-store (12.3.1)

[RUN\_GRADLEW] - expo-splash-screen (0.20.5)

[RUN\_GRADLEW] > Configure project :react-native-reanimated

[PREPARE\_ARTIFACTS] Writing artifacts to *(...)*/bam\_pro\_jach\_jaworski/frontend/build-1702042231017.apk

Build successful

You can find the build artifacts in   
*(...)*/bam\_pro\_jach\_jaworski/frontend/build-1702042231017.apk

Tak utworzoną aplikację można następnie udostępnić do zainstalowania lub przetestować ją na emulatorze instalując ją za pomocą adb np.:

../frontend$: adb install build-1702042231017.apk

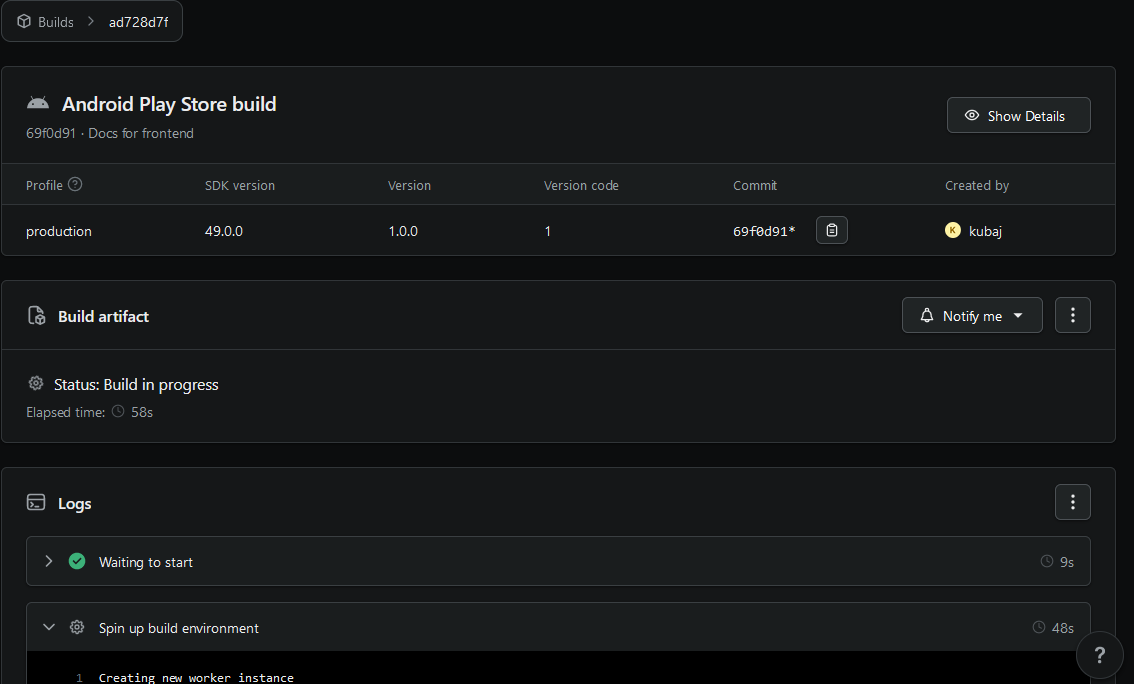
Performing Streamed Install

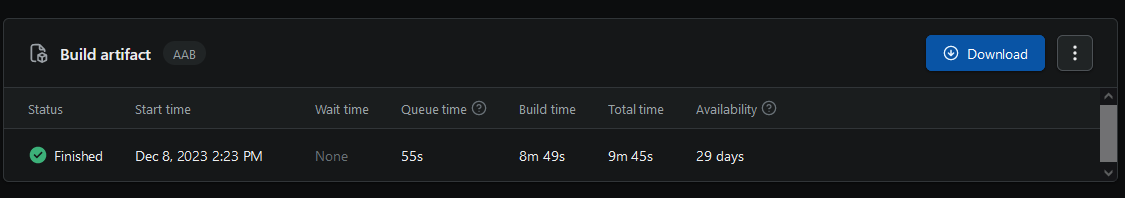
Success

Możliwe jest również zbudowanie aplikacji korzystając z serwerów Expo:



Wybierając build możliwe jest śledzenie postępu:





Po zbudowaniu aplikacji możliwe jest pobranie jej z serwera i zainstalowanie na emulatorze albo telefonie

(...)

Waiting in Free tier queue

|■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■|

√ Build finished

🤖 Android app:

https://expo.dev/artifacts/eas/b7Yp1LZneyWutfyoBDsnKD.apk

√ Install and run the Android build on an emulator? ... yes

√ Successfully downloaded app 1s

Using open emulator: Pixel\_3a

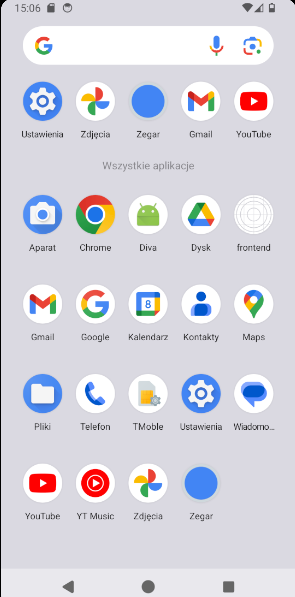
Installing your app...

√ Successfully installed your app!

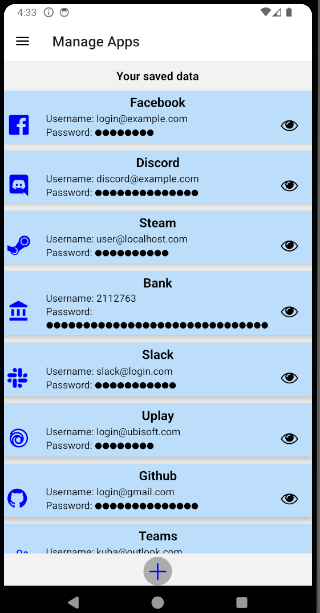
Starting your app...

√ Successfully started your app!

Aplikacja została zainstalowana na emulatorze



Główne okno aplikacji z przykładowymi danymi aplikacji uruchomionej na emulatorze.



Informacje na temat udostępniania aplikacji w sklepie Play dostępne są na podanych stronach:

* <https://developer.android.com/studio/publish>
* <https://docs.expo.dev/submit/introduction/>