

CORDOVA

SPIS TREŚCI

Spis treści	1
Cel zajęć.....	1
Uwaga	1
Instalacja Cordova	2
Utworzenie projektu Cordova i pierwsze uruchomienie.....	2
Podmiana projektu domyślnego na Pogodynę.....	4
Android.....	7
Electron i aplikacja na Windows	9
Debuggowanie aplikacji Electron	11
Dostosowanie aplikacji Electron	11
Budowa aplikacji	13
Commit projektu do GIT.....	14
Podsumowanie.....	14

CEL ZAJĘĆ

Celem głównym zajęć jest zdobycie następujących umiejętności:

- tworzenie hybrydowych aplikacji mobilnych z wykorzystaniem oprogramowania Apache Cordova;
- reużywanie istniejącego kodu HTML do tworzenia aplikacji mobilnych.

W praktycznym wymiarze uczestnicy zamienią swoją aplikację pogodową z LAB D na prostą aplikację mobilną Android.

UWAGA

Ten dokument aktywnie wykorzystuje niestandardowe właściwości. Podobnie jak w LAB A wejdź do **Plik** -> **Informacje** -> **Właściwości** -> **Właściwości zaawansowane** -> **Niestandardowe** i zaktualizuj pola. Następnie uruchom ten dokument ponownie lub **Ctrl+A** -> **F9**.

INSTALACJA CORDOVA

Upewnij się, że w systemie operacyjnym masz poprawnie zainstalowany lub rozpakowany i skonfigurowany menedżer pakietów NPM. Był on już wielokrotnie wykorzystywany, więc powinien być gotowy (por. LAB E, LAB F).

Otwórz ulubiony terminal i zainstaluj oprogramowanie Apache Cordova globalnie dla Twojej instancji NPM, poprzez wykonanie polecenia:

```
> npm install -g cordova
...
added 547 packages in 32s
```

UTWORZENIE PROJEKTU CORDOVA I PIERWSZE URUCHOMIENIE

Utwórz i wejdź poprzez terminal do katalogu `C:\Users\...\Desktop\ai1-cordova`. Następnie utwórz swój projekt z wykorzystaniem polecenia:

```
> cordova create c00000
Creating a new cordova project.
```

Otwórz utworzony katalog `c00000` w Visual Studio Code. Zapoznaj się ze strukturą katalogów.

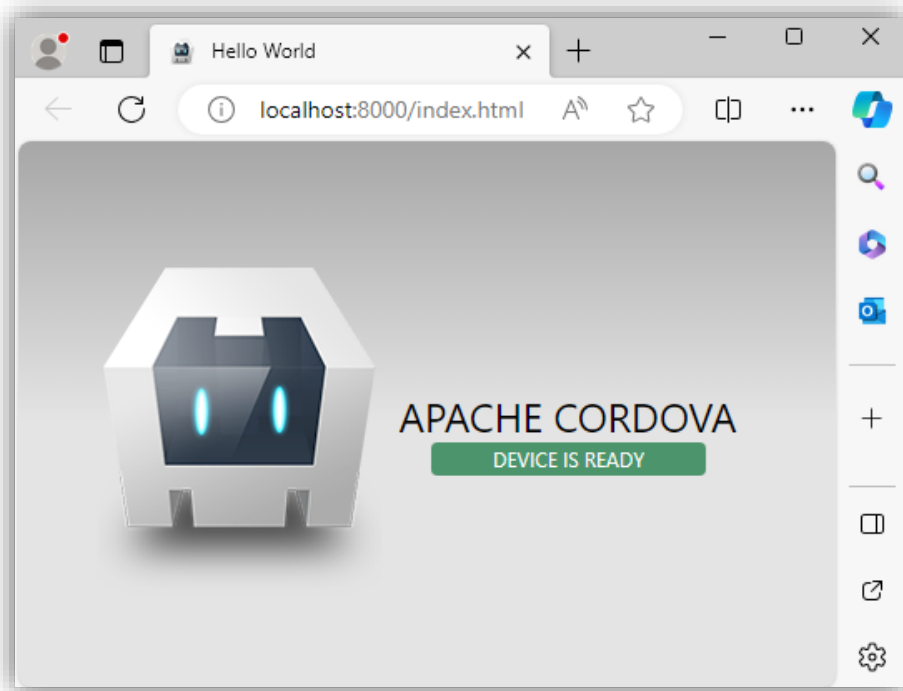
Omów w maksymalnie 100 słowach zawartość projektu na tym etapie:

...maksymalnie 100 słów...

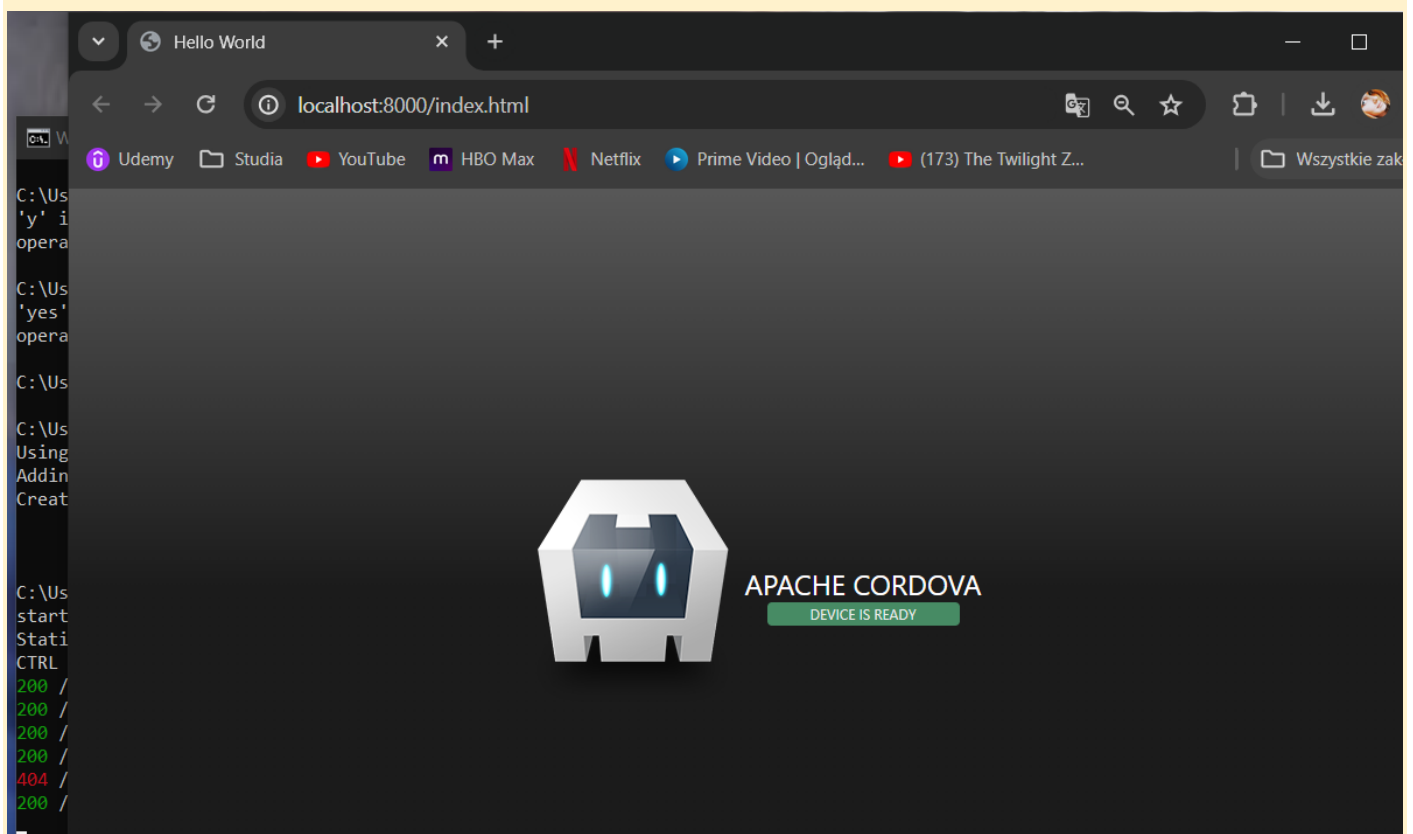
Następnie wróć do terminala, wejdź do katalogu projektu i dodaj platformę browser, po czym uruchom emulację w przeglądarce:

```
> cd c00000
> cordova platform add browser
> cordova emulate browser
```

Powinna uruchomić się przeglądarka z domyślnym projektem:



Wstaw zrzut ekranu emulacji w przeglądarce domyślnego projektu:



Wstaw zrzut ekranu zawartości terminalu po uruchomieniu emulacji i wyświetleniu strony:

Na jakim porcie uruchomiła się Tobie emulacja?

```
C:\Users\Banci>cd ba34753

C:\Users\Banci\ba34753>cordova platform add browser
Using cordova-fetch for cordova-browser
Adding browser project...
Creating Cordova project for cordova-browser:
    Path: C:\Users\Banci\ba34753\platforms\browser
    Name: HelloCordova

C:\Users\Banci\ba34753>cordova emulate browser
startPage = index.html
Static file server running @ http://localhost:8000/index.html
CTRL + C to shut down
200 /index.html (gzip)
200 /css/index.css (gzip)
200 /img/logo.png
200 /cordova.js (gzip)
404 /cordova_plugins.js
200 /js/index.js (gzip)
```

...numer portu...

8000

W maksymalnie 100 słowach opisz zmiany, które zaszły w strukturze katalogów projektu:

W strukturze katalogów projektu zostały wprowadzone zmiany dotyczące dodania katalogu `cordova` w lokalizacji `C:\Users\Banci\AppData\Roaming\npm`. Zmiany te wynikają z instalacji narzędzia `cordova` poprzez `npm`, co jest standardową procedurą w systemie Windows. Katalog ten zawiera pliki i skrypty związane z działaniem `cordova`, takie jak wykonywalne pliki i konfiguracje wymagane do zarządzania aplikacjami mobilnymi.

...maksymalnie 100 słów...

Punkty:	0	1
---------	---	---

PODMIANA PROJEKTU DOMYŚLNEGO NA POGODYNKĘ

Zatrzymaj emulację w przeglądarce, jeśli jest jeszcze uruchomiona, z wykorzystaniem skrótu klawiszowego `Ctrl + C`.

Umieść w katalogu `www` projektu pliki HTML, JS i CSS z klientem REST do wyświetlania pogody, które opracowane zostały w ramach laboratorium LAB D. Upewnij się, że **główny plik HTML ma nazwę `weather.html`**. Upewnij się, że w pliku HTML w dalszym ciągu są poprawne ścieżki względne do plików JS i CSS. Upewnij się, że w kodzie użyty jest **Twój indywidualny klucz** do API.

Edytuj plik `config.xml`. Zmień domyślny plik projektu w znaczniku `<content>` z `index.html` na `weather.html`:

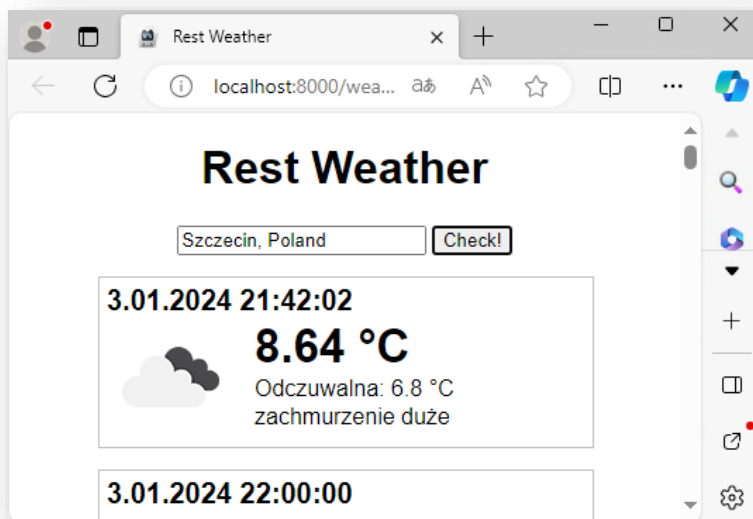
```
<content src="weather.html" />
```

Zmień także nazwę, opis i autora aplikacji w pliku `config.xml`.

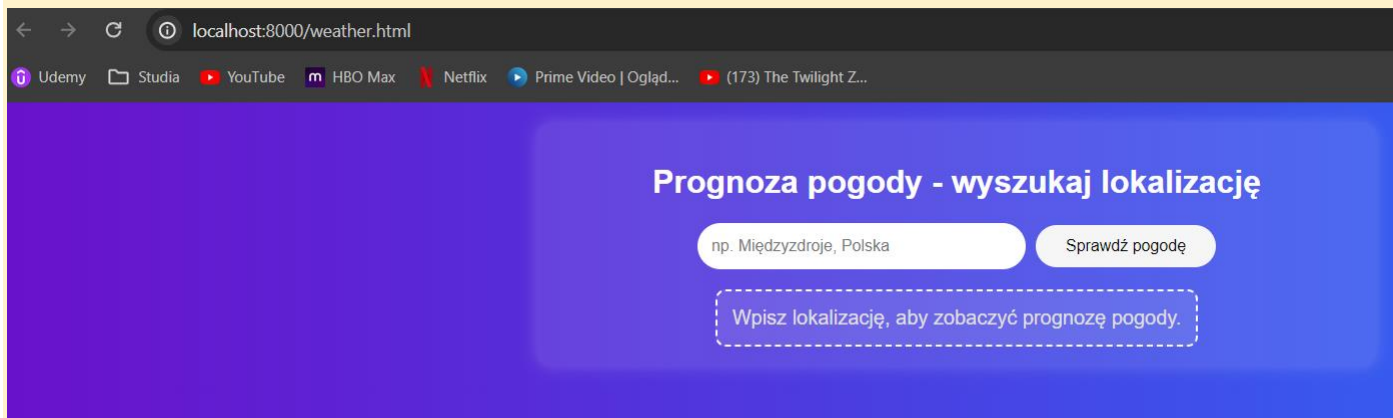
Po wprowadzeniu zmian, zapisz wszystkie pliki i ponownie uruchom emulację projektu w przeglądarce:

```
> cordova emulate browser
```

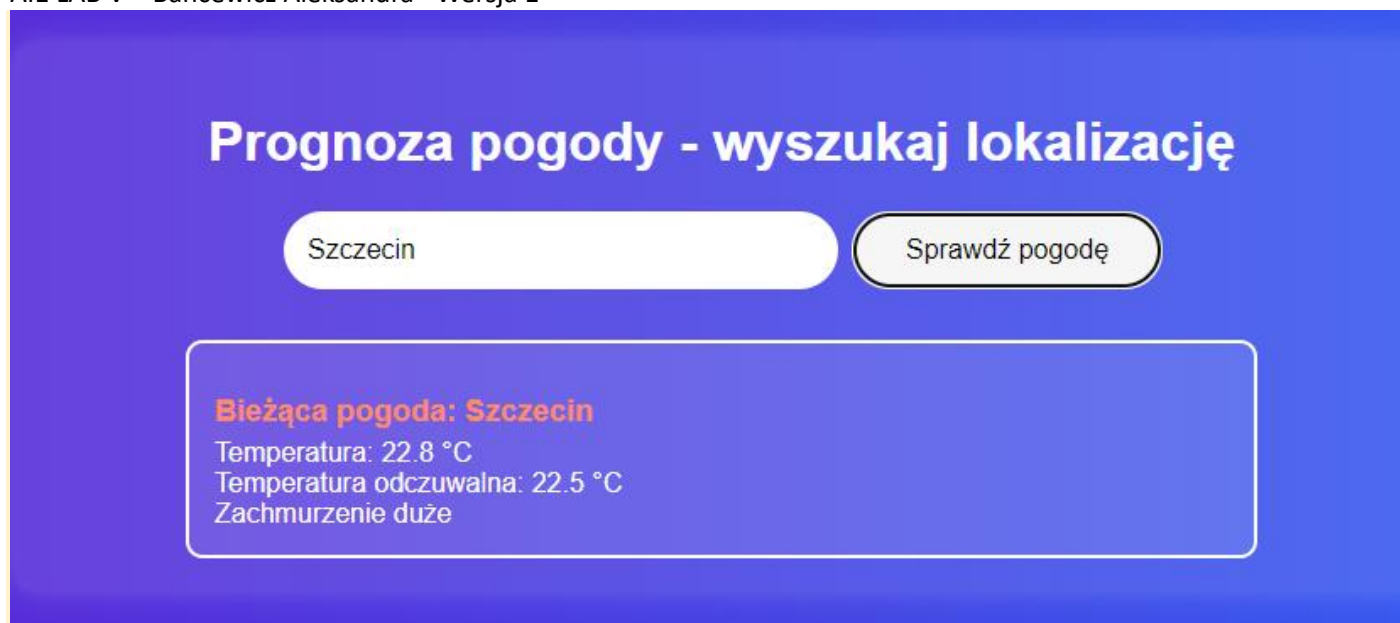
Po uruchomieniu emulacji, w przeglądarce powinna pojawić się aplikacja Pogodynki. Wprowadź nazwę miejscowości i wyszukaj prognozy pogody, przykładowo:



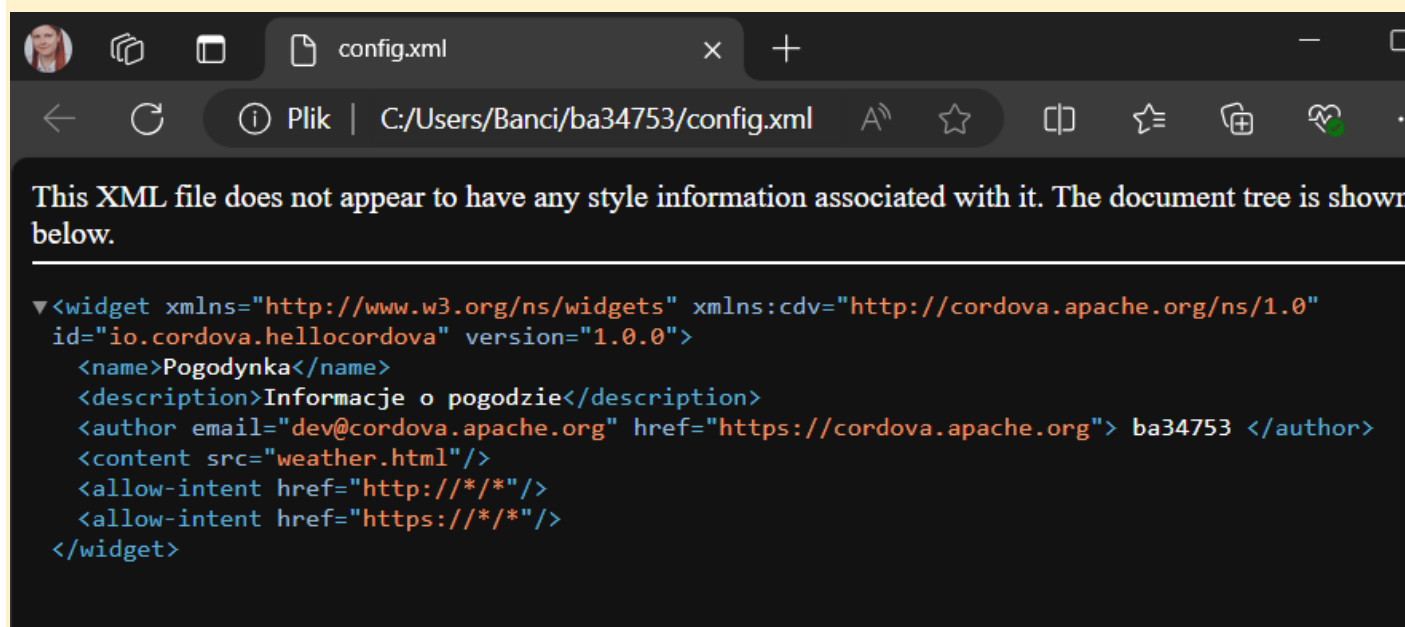
Wstaw zrzut ekranu emulacji w przeglądarce projektu Pogodynki przed wyszukaniem prognoz:



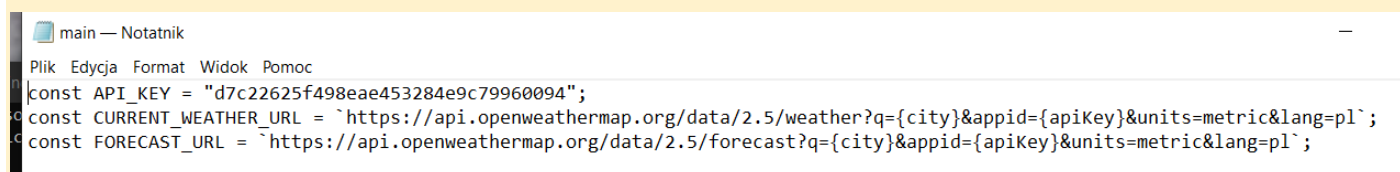
Wstaw zrzut ekranu emulacji pogodynki po wyszukaniu prognoz:



Wstaw zrzut ekranu pliku config.xml:



Wstaw zrzut ekranu fragmentu kodu zawierającego wykorzystanie Twojego klucza do API. Upewnij się, że klucz jest widoczny i że jest to Twój własny klucz:



Punkty:	0	1
---------	---	---

ANDROID

W tej sekcji wykorzystamy projekt Cordova do zbudowania aplikacji Pogodynka na Android. Do budowy wykorzystamy platformę VoltBuilder.

Najpierw w konsoli dodaj platformę Android:

```
> cordova platform add android
```

Następnie utwórz archiwum ZIP z całym katalogiem projektu, tj. `C:\Users\...\Desktop\ai1-cordova\c00000`. Załóżmy, że otrzymane archiwum to `c00000.zip`.

Wejdź na stronę <https://volt.build>. Zarejestruj i zaloguj się. Utworzone zostanie darmowe konto. Przejdź do sekcji Upload, tj. <https://volt.build/upload/>. Wybierz Android i wgraj swoje archiwum ZIP. Poczekaj, aż aplikacja zostanie zbudowana:

Android - The default build engine is now Android 13. This may break your build. [Read this](#) to update your project or learn how to change your build back to the old version.



You're on the Free Plan. Android Debug builds only. [Upgrade here](#)

21:58:22 Processing...

Po kilku minutach aplikacja będzie gotowa do pobrania:

You're on the Free Plan. Android Debug builds only. [Upgrade here](#)




Cordova Weather App c00000

Android debug

Wed Jan 03 2024 21:59:14 GMT+0100 (czas środkowoeuropejski standardowy)

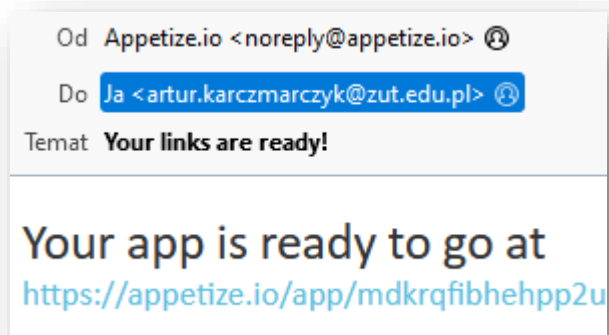
 Download

 Copy URL for use on device to clipboard

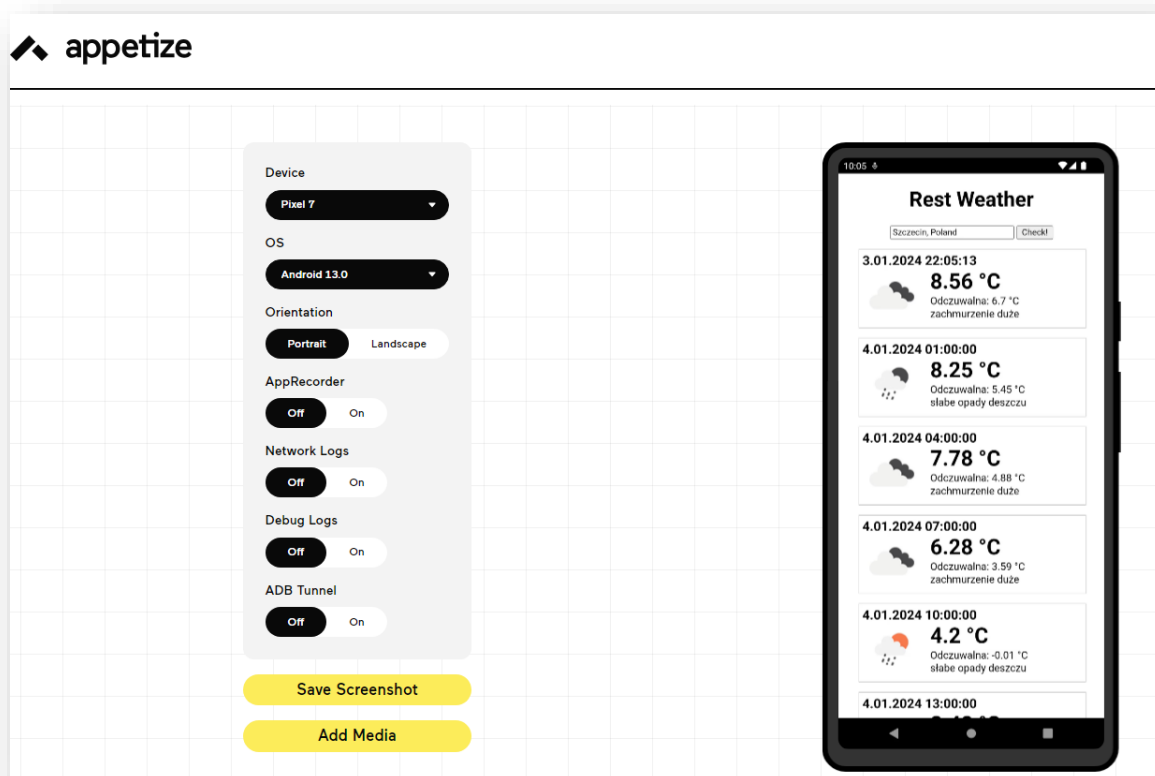
 View Log

Pobierz aplikację. Będzie to plik `.apk`.

Teraz jesteśmy gotowi do przetestowania działania aplikacji w emulatorze Android. Wykorzystamy do tego celu usługę appetize. Wejdź na stronę <https://appetize.io/>. Przejdź do **Demo**. Przyciśnij **Upload your app** i wybierz z dysku swoją aplikację. Podaj adres email i kliknij **Generate Links**. Adres może być uczelniany lub prywatny. Testowane na @zut.edu.pl i @gmail.com. W ciągu kilku chwil otrzymasz na mail link do aplikacji:



Przejdź pod podany link w przeglądarce, wybierz urządzenie, system operacyjny i uruchom emulację:



Wstaw zrzut ekranu Twojej aplikacji w **appetize** przed wyszukaniem prognozy:



Wstaw zrzut ekranu Twojej aplikacji w **appetize** po wyszukaniu prognozy:

Wstaw zrzut ekranu maila od **appetize** z linkiem do aplikacji:

Wstaw link do aplikacji w **appetize**:

<https://appetize.io/app/.....>

Wstaw zrzut ekranu fragmentu strony **VoltBuilder** przedstawiający zbudowaną aplikację:

Punkty:	0	1
---------	---	---

Rada!

Jeśli masz urządzenie Android, możesz zeskanować telefonem kod QR wyświetlony w **VoltBuilder**. Pod podanym adresem znajdziesz aplikację **.apk** do pobrania na Twój telefon. Następnie możesz zainstalować tę aplikację w telefonie i korzystać na co dzień!

ELECTRON I APLIKACJA NA WINDOWS

Dodaj do projektu platformę electron:

```
> cordova platform add electron
```

Przejdź do Visual Studio Code. W głównym katalogu projektu utwórz plik **build.json** o następującej zawartości:

```
{  
  "electron": {
```

```
"windows": {  
  "package": [  
    "zip"  
  ]  
}  
}
```

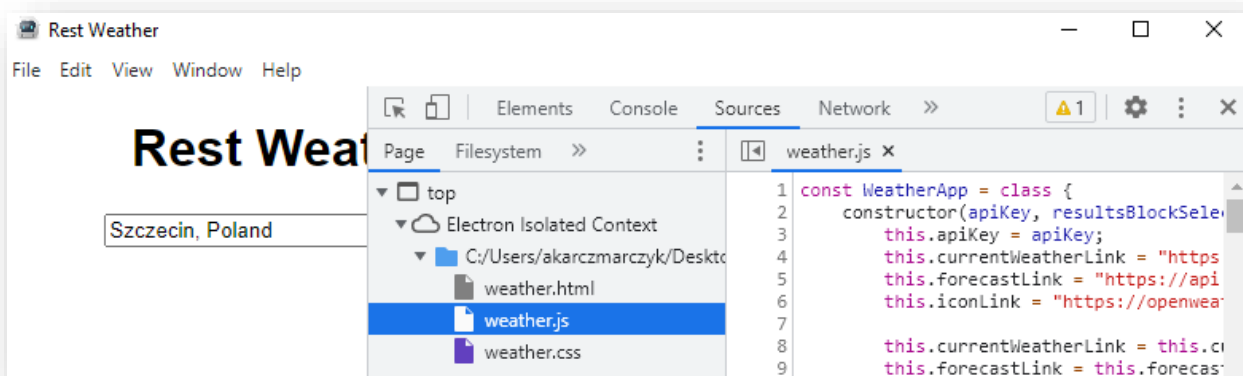
Przy budowie aplikacji, biblioteka Electron tworzyć będzie archiwum ZIP z projektem, zamiast instalatora. Dostępne postaci pakietów instalacyjnych opisane są tutaj:

<https://cordova.apache.org/docs/en/12.x/guide/platforms/electron/index.html#adding-a-package>.

Uruchom teraz emulację aplikacji Electron:

```
> cordova emulate electron -nobuild
```

W efekcie uruchomi się aplikacja Electron:



Wstaw zrzut ekranu uruchomionej aplikacji Electron:



Zamknij narzędzia deweloperskie (panel z prawej strony) i wyszukaj prognozę pogody. Wstaw zrzut ekranu:



Punkty:	0	1
---------	---	---

DEBUGGOWANIE APLIKACJI ELECTRON

Ponownie włącz emulację aplikacji electron. W narzędziach deweloperskich (panel z prawej strony), wybierz zakładkę „Sources”. Znajdź kod JS odpowiedzialny za pobranie bieżącej prognozy pogody i ustaw pułapkę (ang. breakpoint) na w linii odpowiedniej funkcji. Wyszukaj prognozę pogody. Upewnij się, że pułapka się aktywowała.

Wstaw zrzut ekranu całości okienka electron z aplikacją zatrzymaną na breakpointie:



Upewnij się, że widoczne są:

- nad treścią aplikacji informacja, że wykonanie zostało wstrzymane w debuggerze
- debugger zatrzymany na pułapce (linia podświetlona na niebiesko)
- w okienku Scope widoczne są wartości zmiennych

Punkty:	0	1
---------	---	---

DOSTOSOWANIE APLIKACJI ELECTRON

W Visual Studio Code utwórz w projekcie podkatalog `res\electron`. Utwórz w dowolnym programie graficznym ikonkę aplikacji o wymiarach 512x512 pikseli (format PNG). Utworzony plik zapisz w katalogu `res\electron`, a następnie w pliku `config.xml` skonfiguruj ikonę dla platformy electron, przykładowo:

```
<platform name="electron">
  <icon src="res/electron/cloud-sun.png" />
</platform>
```

Następnie utwórz plik `res\electron\settings.json` i zarejestruj go w `config.xml`:

```
<platform name="electron">
  <icon src="res/electron/cloud-sun.png" />
  <preference name="ElectronSettingsFilePath" value="res/electron/settings.json" />
</platform>
```

W pliku `settings.json` dostosuj szerokość i wysokość aplikacji, a także ukryj menu. Dokumentacja:

<https://cordova.apache.org/docs/en/12.x/guide/platforms/electron/index.html#customizing-the-application's-window-options>

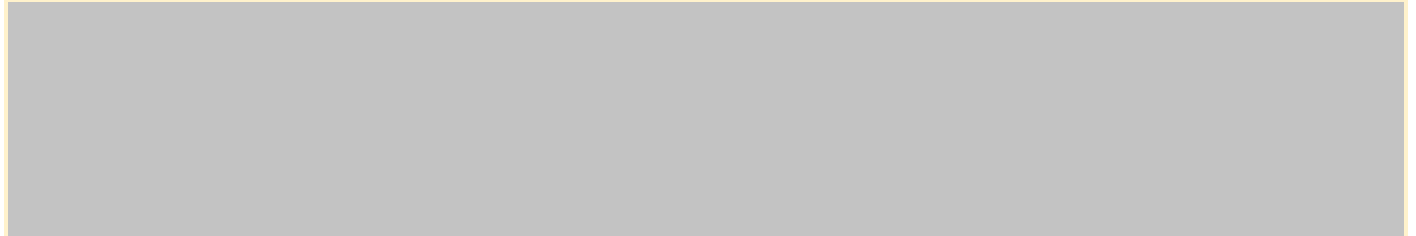
Przykładowa zawartość pliku `settings.json`:

```
{
  "browserWindow": {
```

```
{  
  "width": 1024,  
  "height": 600,  
  "autoHideMenuBar": true  
}
```

Ponownie dokonaj emulacji aplikacji, tym razem z przełącznikiem `--release`, który ukryje narzędzia deweloperskie.

Wstaw zrzut ekranu kodu pliku `settings.json`:



Wstaw zrzut ekranu kodu pliku `config.xml`:



Wstaw zrzut ekranu pliku PNG z ikoną o wymiarach `512x512px`:



Wstaw zrzut ekranu uruchomionej aplikacji. Upewnij się, że wysokość i szerokość odpowiada wartościom z konfiguracji, pasek menu jest ukryty, ikonka jest zgodna ze skonfigurowaną, a narzędzia deweloperskie są zamknięte:



Punkty:	0	1
---------	---	---

BUDOWA APLIKACJI

Wykonaj polecenie budowy aplikacji Electron:

```
> cordova build electron --release
```

Oczekaj kilka minut. Po zakończonym procesie wynikowe archiwum ZIP znajdzie się w katalogu `\platforms\electron\build`.

Skopiuj archiwum do innego katalogu, rozpakuj, wejdź do środka rozpakowanego katalogu i uruchom plik EXE.

Wstaw zrzut ekranu zawartości katalogu `platforms\electron\build`:



Wstaw zrzut ekranu zawartości rozpakowanego archiwum ZIP z aplikacją:



Wstaw zrzut ekranu uruchomionej aplikacji przed sprawdzeniem pogody:



Wstaw zrzut ekranu uruchomionej aplikacji po sprawdzeniu prognozy pogody:



Wstaw zrzut ekranu eksploratora plików z zaznaczonym plikiem EXE aplikacji z ikonką zgodną z konfiguracją:

Punkty:	0	1
---------	---	---

COMMIT PROJEKTU DO GIT

Zacommittuj i pushnij swoje rozwiązanie do **swojego** repozytorium GIT. **UWAGA! Nie commitować binarek!!!**

Upewnij się, czy wszystko dobrze się wysłało. Jeśli tak, to z poziomu przeglądarki utwórz branch o nazwie `lab-v` na podstawie głównej gałęzi kodu.

Podaj link do brancha `lab-v` w swoim repozytorium:

...link, np. <https://github.com/inazwisko/ai1-lab/tree/lab-v...>

PODSUMOWANIE

W kilku zdaniach podsumuj zdobyte podczas tego laboratorium umiejętności.

...podsumowanie...

Zweryfikuj kompletność sprawozdania. Utwórz PDF i wyślij w terminie.