

## 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 .....	10
Debuggowanie aplikacji Electron .....	12
Dostosowanie aplikacji Electron .....	13
Budowa aplikacji .....	15
Commit projektu do GIT.....	17
Podsumowanie.....	18

### 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 c50970
Creating a new cordova project.
```

Otwórz utworzony katalog `c50970` w Visual Studio Code. Zapoznaj się ze strukturą katalogów.

Omów w maksymalnie 100 słowach zawartość projektu na tym etapie:

W utworzonym projekcie mamy następujące pliki

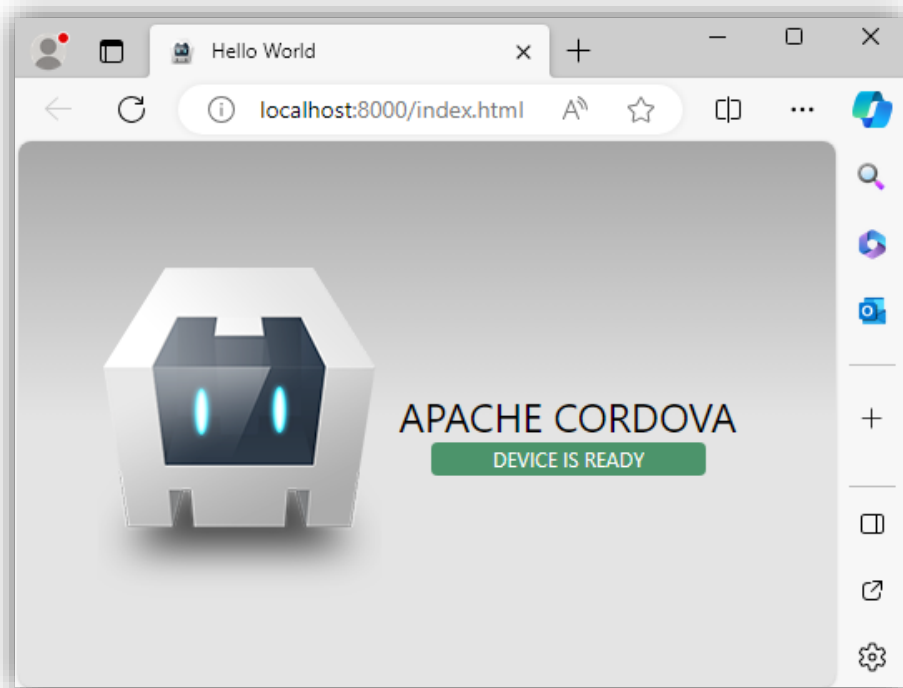
Index.css Index.js Index.html logo.png config.xml I package.json

...maksymalnie 100 słów...

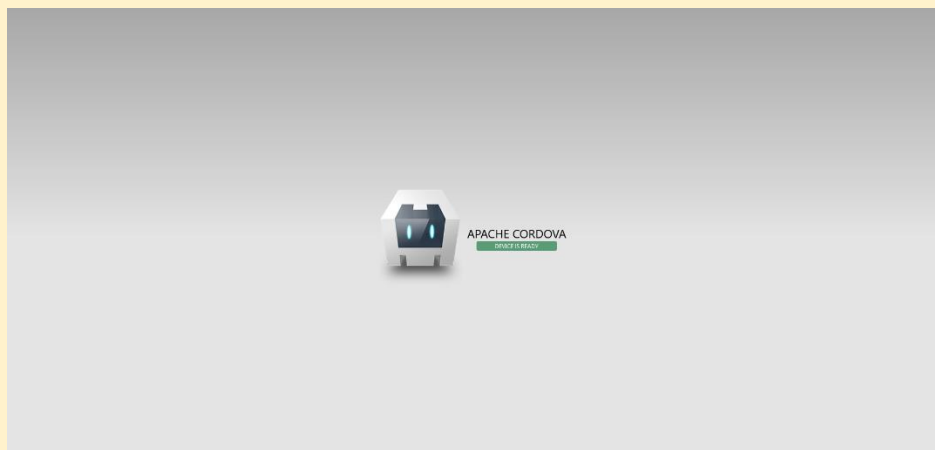
Następnie wróć do terminala, wejdź do katalogu projektu i dodaj platformę browser, po czym uruchom emulację w przeglądarce:

```
> cd c50970
> 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:

```
C:\IT\Studia\Aplikacje Internetowe 1\AI1LAB1\c50970>cordova platform add browser
Using cordova-fetch for cordova-browser
Adding browser project...
Creating Cordova project for cordova-browser:
  Path: C:\IT\Studia\Aplikacje Internetowe 1\AI1LAB1\c50970\platforms\browser
  Name: HelloCordova

C:\IT\Studia\Aplikacje Internetowe 1\AI1LAB1\c50970>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 /js/index.js (gzip)
200 /cordova.js (gzip)
404 /cordova_plugins.js
200 /img/logo.png
200 /favicon.ico (gzip)
```

Na jakim porcie uruchomiła się Tobie emulacja?

...numer portu = 8000

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

Dodały się moduły node oraz katalog platforms zawierający browser

Mamy też pliki package.json oraz package-lock.json

...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:

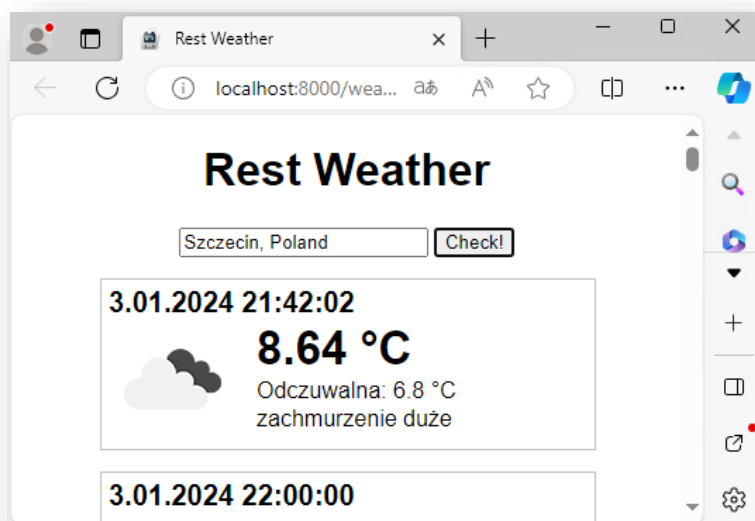
```
<content src="weather.html" />
```

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:

PREGNANT WOMEN WITH SUBSTANTIAL RISK	
1. $\frac{1}{2}$ (100%)	2. $\frac{1}{2}$ (100%)
3. $\frac{1}{2}$ (100%)	4. $\frac{1}{2}$ (100%)
5. $\frac{1}{2}$ (100%)	6. $\frac{1}{2}$ (100%)
7. $\frac{1}{2}$ (100%)	8. $\frac{1}{2}$ (100%)
9. $\frac{1}{2}$ (100%)	10. $\frac{1}{2}$ (100%)
11. $\frac{1}{2}$ (100%)	12. $\frac{1}{2}$ (100%)
13. $\frac{1}{2}$ (100%)	14. $\frac{1}{2}$ (100%)
15. $\frac{1}{2}$ (100%)	16. $\frac{1}{2}$ (100%)
17. $\frac{1}{2}$ (100%)	18. $\frac{1}{2}$ (100%)
19. $\frac{1}{2}$ (100%)	20. $\frac{1}{2}$ (100%)
21. $\frac{1}{2}$ (100%)	22. $\frac{1}{2}$ (100%)
23. $\frac{1}{2}$ (100%)	24. $\frac{1}{2}$ (100%)
25. $\frac{1}{2}$ (100%)	26. $\frac{1}{2}$ (100%)
27. $\frac{1}{2}$ (100%)	28. $\frac{1}{2}$ (100%)
29. $\frac{1}{2}$ (100%)	30. $\frac{1}{2}$ (100%)
31. $\frac{1}{2}$ (100%)	32. $\frac{1}{2}$ (100%)
33. $\frac{1}{2}$ (100%)	34. $\frac{1}{2}$ (100%)
35. $\frac{1}{2}$ (100%)	36. $\frac{1}{2}$ (100%)
37. $\frac{1}{2}$ (100%)	38. $\frac{1}{2}$ (100%)
39. $\frac{1}{2}$ (100%)	40. $\frac{1}{2}$ (100%)
41. $\frac{1}{2}$ (100%)	42. $\frac{1}{2}$ (100%)
43. $\frac{1}{2}$ (100%)	44. $\frac{1}{2}$ (100%)
45. $\frac{1}{2}$ (100%)	46. $\frac{1}{2}$ (100%)
47. $\frac{1}{2}$ (100%)	48. $\frac{1}{2}$ (100%)
49. $\frac{1}{2}$ (100%)	50. $\frac{1}{2}$ (100%)
51. $\frac{1}{2}$ (100%)	52. $\frac{1}{2}$ (100%)
53. $\frac{1}{2}$ (100%)	54. $\frac{1}{2}$ (100%)
55. $\frac{1}{2}$ (100%)	56. $\frac{1}{2}$ (100%)
57. $\frac{1}{2}$ (100%)	58. $\frac{1}{2}$ (100%)
59. $\frac{1}{2}$ (100%)	60. $\frac{1}{2}$ (100%)
61. $\frac{1}{2}$ (100%)	62. $\frac{1}{2}$ (100%)
63. $\frac{1}{2}$ (100%)	64. $\frac{1}{2}$ (100%)
65. $\frac{1}{2}$ (100%)	66. $\frac{1}{2}$ (100%)
67. $\frac{1}{2}$ (100%)	68. $\frac{1}{2}$ (100%)
69. $\frac{1}{2}$ (100%)	70. $\frac{1}{2}$ (100%)
71. $\frac{1}{2}$ (100%)	72. $\frac{1}{2}$ (100%)
73. $\frac{1}{2}$ (100%)	74. $\frac{1}{2}$ (100%)
75. $\frac{1}{2}$ (100%)	76. $\frac{1}{2}$ (100%)
77. $\frac{1}{2}$ (100%)	78. $\frac{1}{2}$ (100%)
79. $\frac{1}{2}$ (100%)	80. $\frac{1}{2}$ (100%)
81. $\frac{1}{2}$ (100%)	82. $\frac{1}{2}$ (100%)
83. $\frac{1}{2}$ (100%)	84. $\frac{1}{2}$ (100%)
85. $\frac{1}{2}$ (100%)	86. $\frac{1}{2}$ (100%)
87. $\frac{1}{2}$ (100%)	88. $\frac{1}{2}$ (100%)
89. $\frac{1}{2}$ (100%)	90. $\frac{1}{2}$ (100%)
91. $\frac{1}{2}$ (100%)	92. $\frac{1}{2}$ (100%)
93. $\frac{1}{2}$ (100%)	94. $\frac{1}{2}$ (100%)
95. $\frac{1}{2}$ (100%)	96. $\frac{1}{2}$ (100%)
97. $\frac{1}{2}$ (100%)	98. $\frac{1}{2}$ (100%)
99. $\frac{1}{2}$ (100%)	100. $\frac{1}{2}$ (100%)

Wstaw zrzut ekranu pliku config.xml:

```

1  <?xml version="1.0" encoding="utf-8"?>
2  <widget id="io.cordova.hellocordova" version="1.0.0" xmlns="http://www.w3.org/ns/widgets" xmlns:cdv="http://cordova.apache.org/ns/1.0">
3      <name>HelloCordova</name>
4      <description>Sample Apache Cordova App</description>
5      <author email="dev@cordova.apache.org" href="https://cordova.apache.org">
6          Apache Cordova Team
7      </author>
8      <content src="weather.html" />
9      <allow-intent href="http:///*/*" />
10     <allow-intent href="https:///*/*" />
11 </widget>
12

```

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

```
const apiKey : string = "24fe27b9cf67b55095f9a18ee2f4c351";

+ usages new *

function findLocation() : void {
    let locationData = document.getElementById("address").value.split(" ");
    document.getElementById(
        elementid: "city"
    ).innerHTML = "Pogoda w mieście ${locationData[1]}";
    document.getElementById( elementid: "weatherContainer").innerHTML = "";
    let location : Promise<Response> = fetch(
        inputs: `http://api.openweathermap.org/geo/1.0/direct?q=${locationData[1]},${locationData[0]}&limit=1&appid=${apiKey}`
    );
}
```

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

## ANDROID

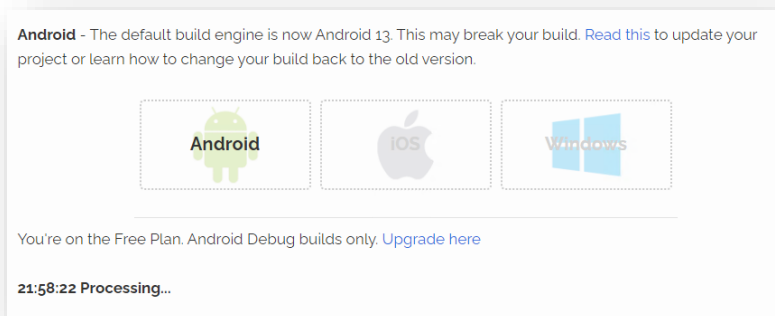
W tej sekcji wykorzystamy projekt Cordova do zbudowania aplikacji Pogodynka na Android. Do budowy wykorzystamy platformę VoltBuilder.

Najpierw w konsoli dodaj platformę Android:

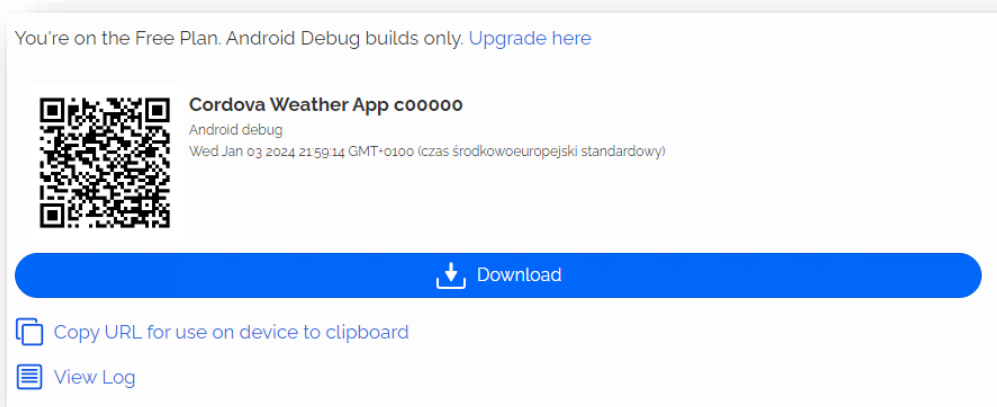
```
> cordova platform add android
```

Następnie utwórz archiwum ZIP z całym katalogiem projektu, tj. `C:\Users\...\Desktop\ai1-cordova\c50970`. Załóżmy, że otrzymane archiwum to `c50970.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:

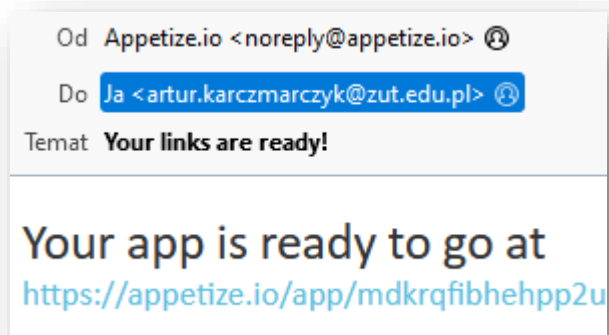


Po kilku minutach aplikacja będzie gotowa do pobrania:

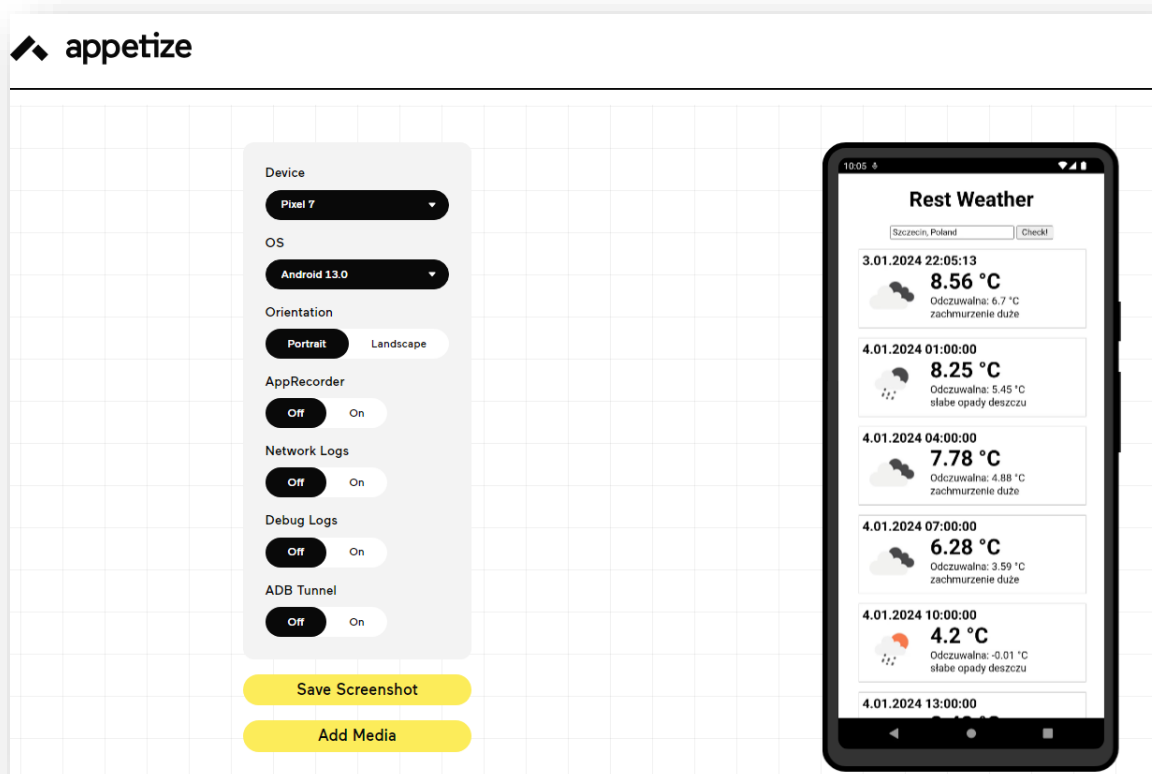


Pobierz aplikację. Będzie to plik `.apk`.

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:

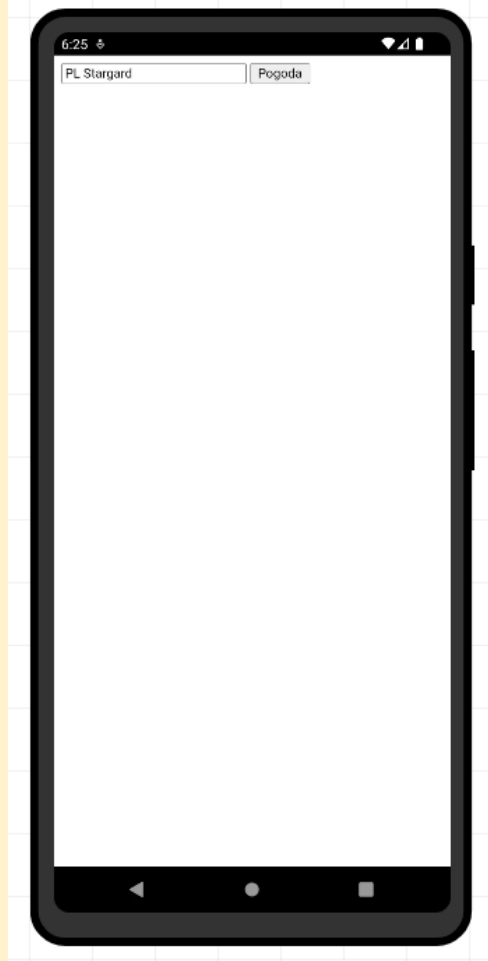


Przejdź pod podany link w przeglądarce, wybierz urządzenie, system operacyjny i uruchom emulację:

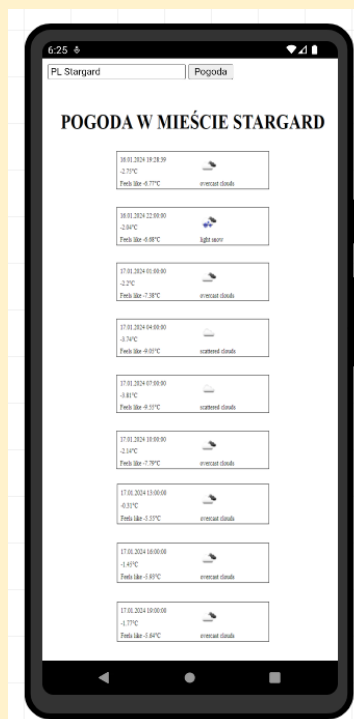


Wstaw zrzut ekranu Twojej aplikacji w **appetize** przed wyszukaniem prognozy:

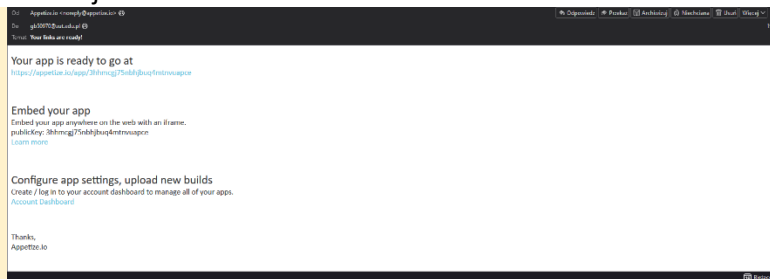




Wstaw zrzut ekranu Twojej aplikacji w **appetize** po wyszukaniu prognozy:



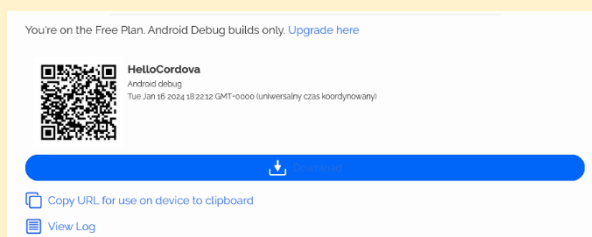
Wstaw zrzut ekranu maila od **appetize** z linkiem do aplikacji:



Wstaw link do aplikacji w **appetize**:

<https://appetize.io/app/3hhmcgj75nbhjbug4mtnvuapce?device=pixel7&osVersion=13.0>

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"
      ]
    }
  }
}
```

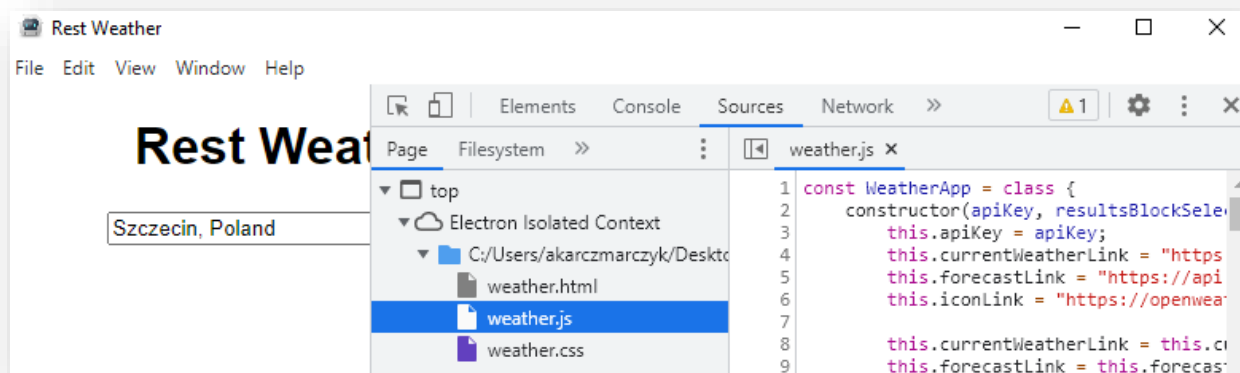
Przy budowie aplikacji, biblioteka Electron tworzyć będzie archiwum ZIP z projektem, zamiast instalatora. Dostępne postaci pakietów instalacyjnych opisane są tutaj:

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

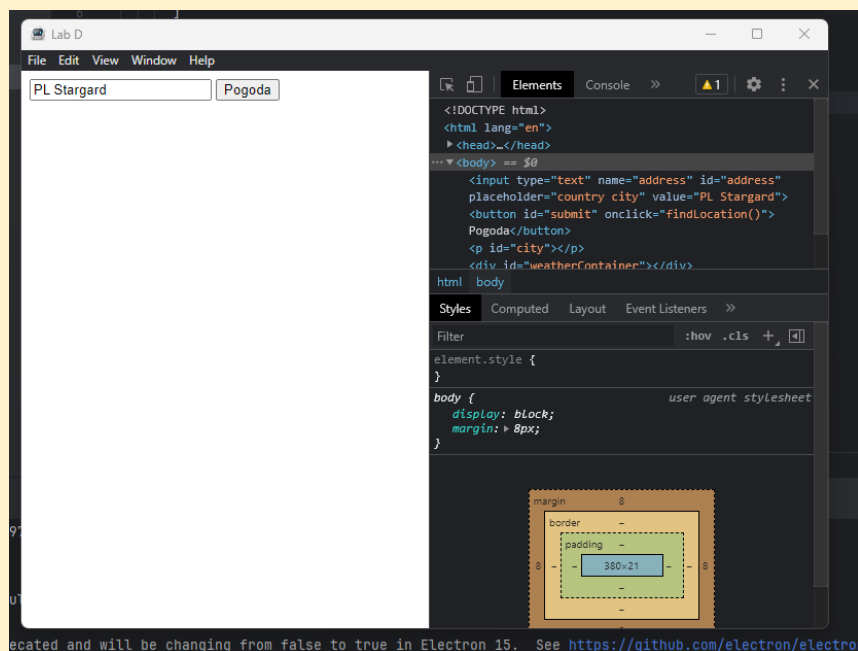
Uruchom teraz emulację aplikacji Electron:

```
> 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:

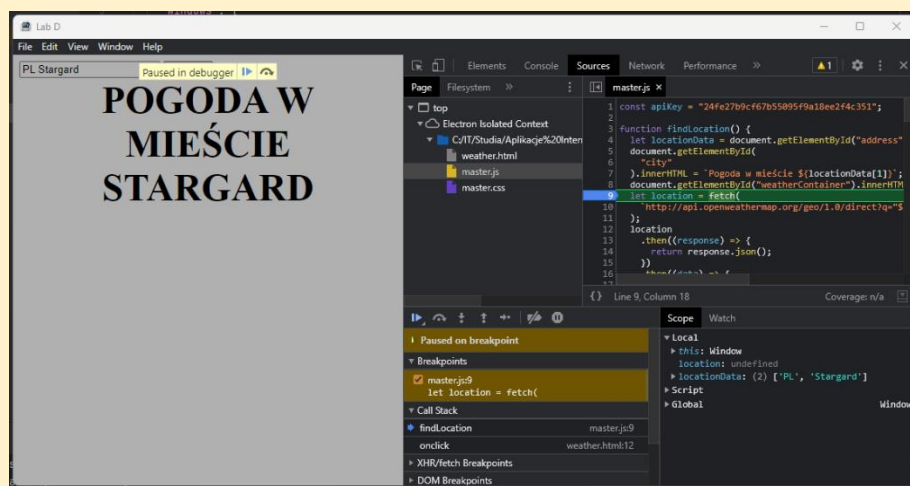
0

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 breakpointcie:



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:

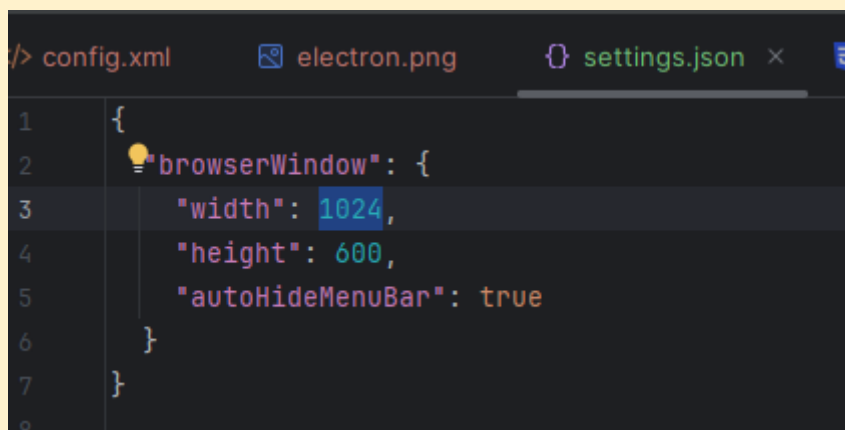
<https://cordova.apache.org/docs/en/12.x/guide/platforms/electron/index.html#customizing-the-application's-window-options>

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`:



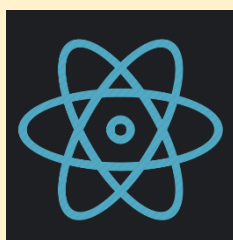
Wstaw zrzut ekranu kodu pliku `config.xml`:

```

1  <?xml version='1.0' encoding='utf-8'?>
2  <widget id="io.cordova.hellocordova" version="1.0.0" xmlns="http://www.w3.org/ns/widgets" xmlns:cdv="http://cordova.apache.org/ns/1.0">
3      <name>HelloCordova</name>
4      <description>Sample Apache Cordova App</description>
5      <author email="dev@cordova.apache.org" href="https://cordova.apache.org">
6          Apache Cordova Team
7      </author>
8      <content src="weather.html" />
9      <allow-intent href="http://*/*" />
10     <allow-intent href="https://*/*" />
11     <platform name="electron">
12         <icon src="res/electron/electron.png" />
13         <preference name="ElectronSettingsFilePath" value="res/electron/settings.json" />
14     </platform>
15 </widget>
16

```

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

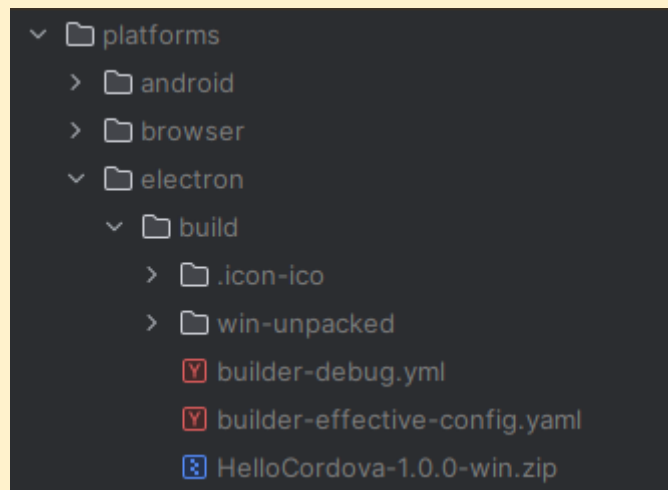
Wykonaj polecenie budowy aplikacji Electron:

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

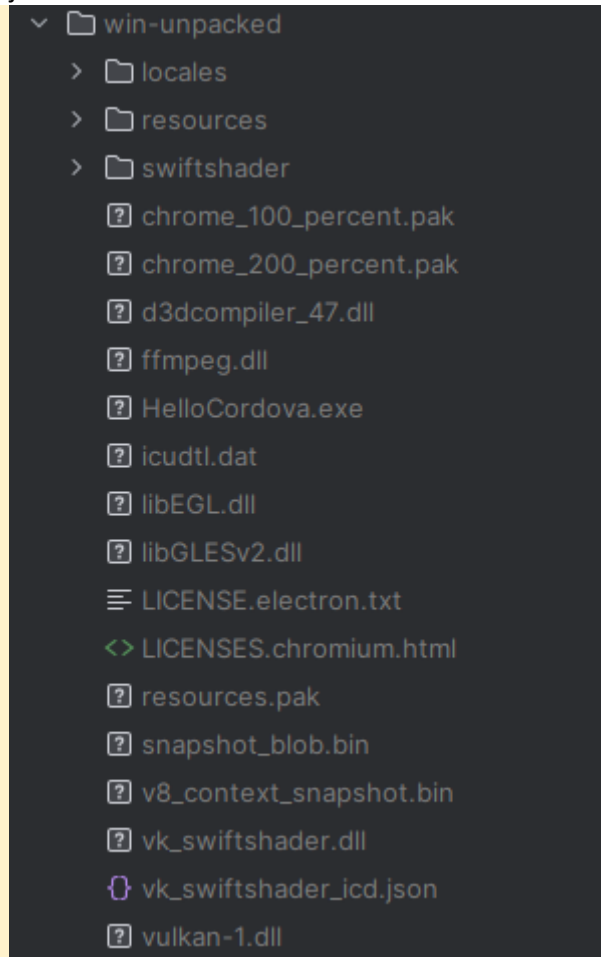
Oczekaj kilka minut. Po zakończonym procesie wynikowe archiwum ZIP znajdzie się w katalogu `\platforms\electron\build`.

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

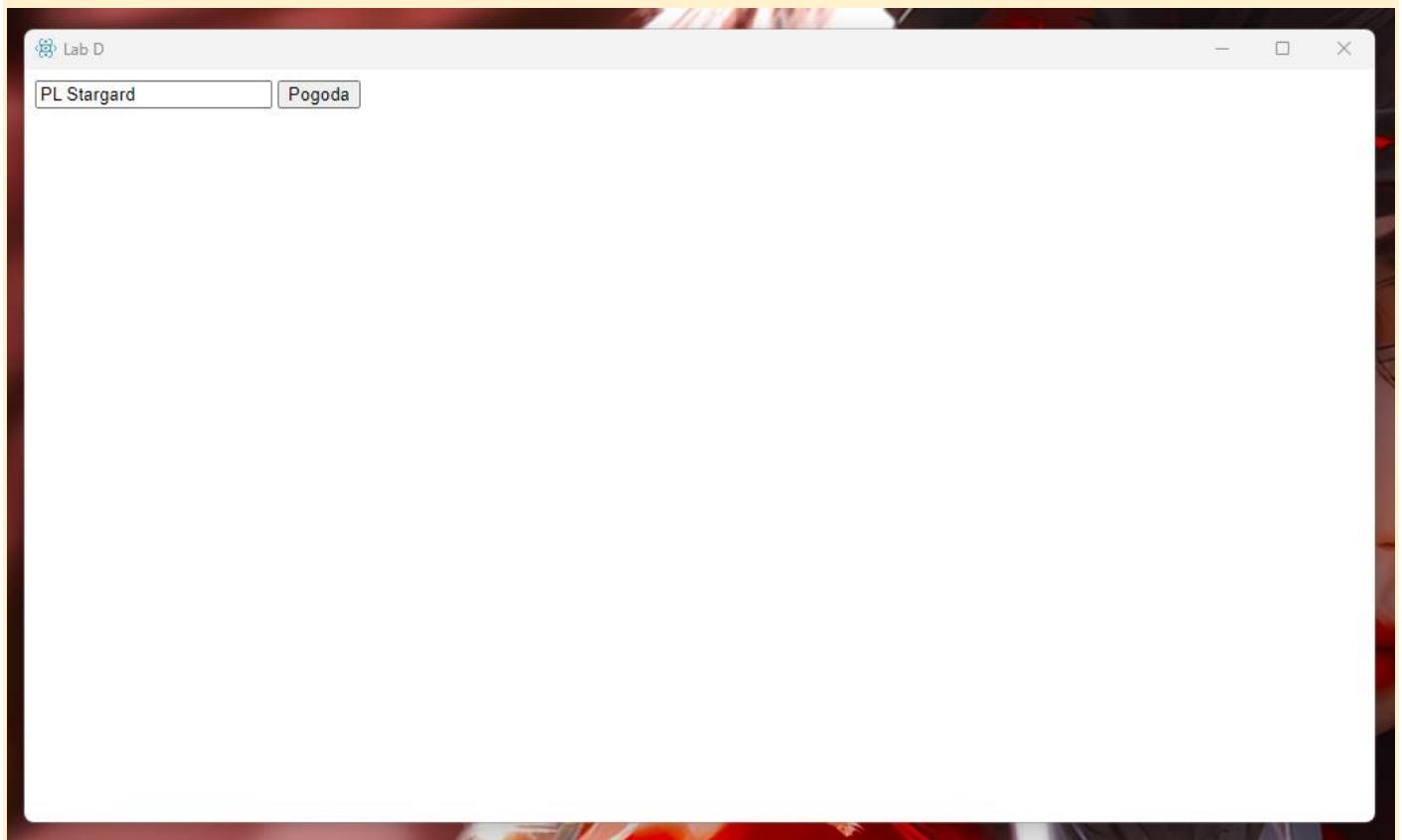
Wstaw zrzut ekranu zawartości katalogu `platforms\electron\build`:



Wstaw zrzut ekranu zawartości rozpakowanego archiwum ZIP z aplikacją:



Wstaw zrzut ekranu uruchomionej aplikacji przed sprawdzeniem pogody:

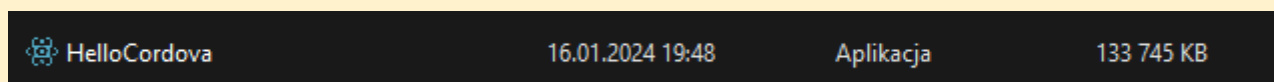




Wstaw zrzut ekranu uruchomionej aplikacji po sprawdzeniu prognozy pogody:



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



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

## COMMIT PROJEKTU DO GIT

Zacommituj 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/1Batrex1/ai1-lab1/tree/lab-v>

## PODSUMOWANIE

W kilku zdaniach podsumuj zdobyte podczas tego laboratorium umiejętności.

Nauczyłem się że w sumie cordova to fajne środowisko jeśli chce się wydać aplikację na wiele platform

...podsumowanie...

Zweryfikuj kompletność sprawozdania. Utwórz PDF i wyślij w terminie.