

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.....	5
Android.....	8
Electron i aplikacja na Windows	13
Debuggowanie aplikacji Electron	15
Dostosowanie aplikacji Electron	16
Budowa aplikacji	20
Commit projektu do GIT.....	24
Podsumowanie.....	24

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 c53757
Creating a new cordova project.
```

Otwórz utworzony katalog `c53757` w Visual Studio Code. Zapoznaj się ze strukturą katalogów.

Omów w maksymalnie 100 słowach zawartość projektu na tym etapie:

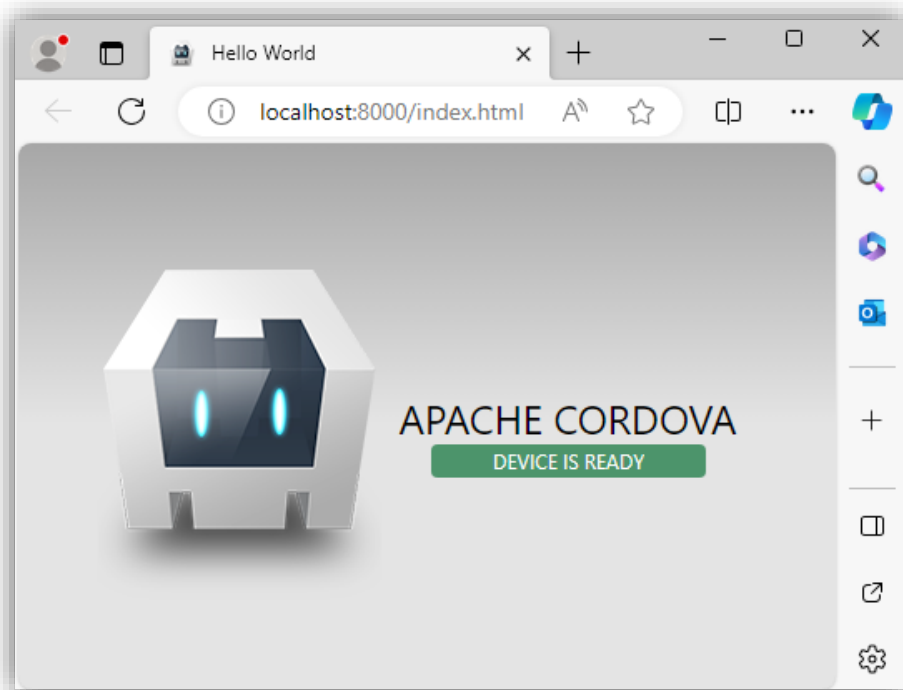
Projekt zawiera plik css, js, html a także xml i json

...maksymalnie 100 słów...

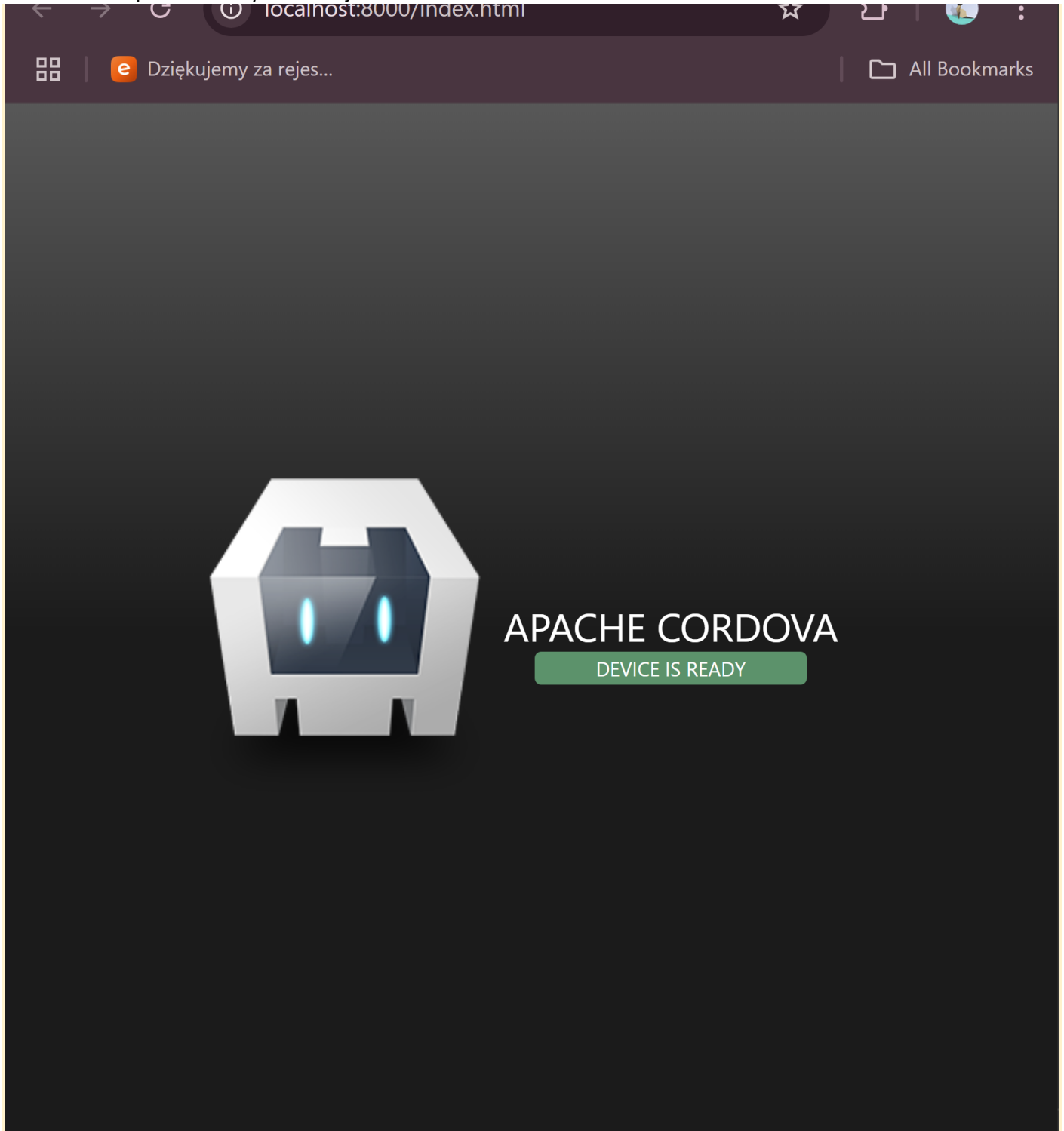
Następnie wróć do terminala, wejdź do katalogu projektu i dodaj platformę browser, po czym uruchom emulację w przeglądarce:

```
> cd c53757
> 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:\Users\kopan\OneDrive\Desktop\judys\aplikacje_internetowe\cordova\c53757>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 /cordova.js (gzip)
200 /img/logo.png
404 /cordova_plugins.js
200 /js/index.js (gzip)
200 /favicon.ico (gzip)
```

Na jakim porcie uruchomiła się Tobie emulacja?

8000

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

Dodano nowy plik json i podkatalog node_modules i platforms\browser

...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:

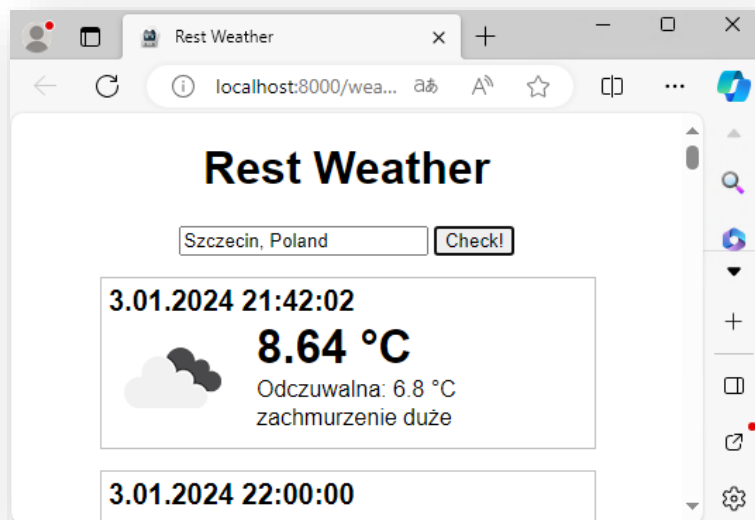
```
<content src="weather.html" />
```

Zmień także identyfikator, nazwę, opis i autora aplikacji w pliku **config.xml**. Nazwa aplikacji musi składać się z Twojego imienia i numeru indeksu. Identyfikatorem musi być odwrócona nazwa domenowa WI wraz z Twoim identyfikatorem LDAP. Przykładowo:

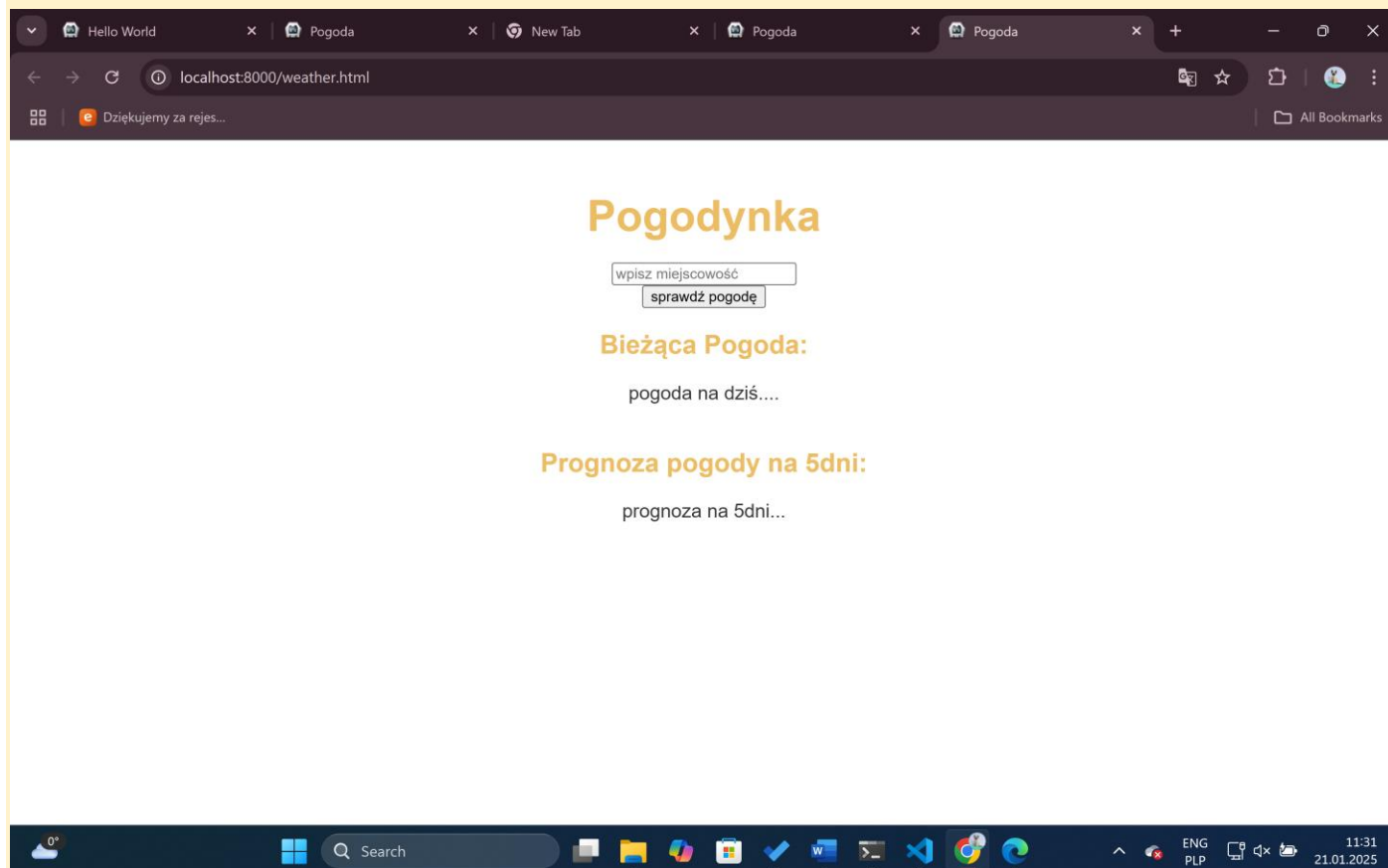
```
<widget id="pl.edu.zut.wi.ni53757" ..... >
<name>Judyta 53757</name>
<description>Weather App</description>
<author email="kj53757@zut.edu.pl" href="https://www.wi.zut.edu.pl">
  Artur Karczmarczyk
</author>
```

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

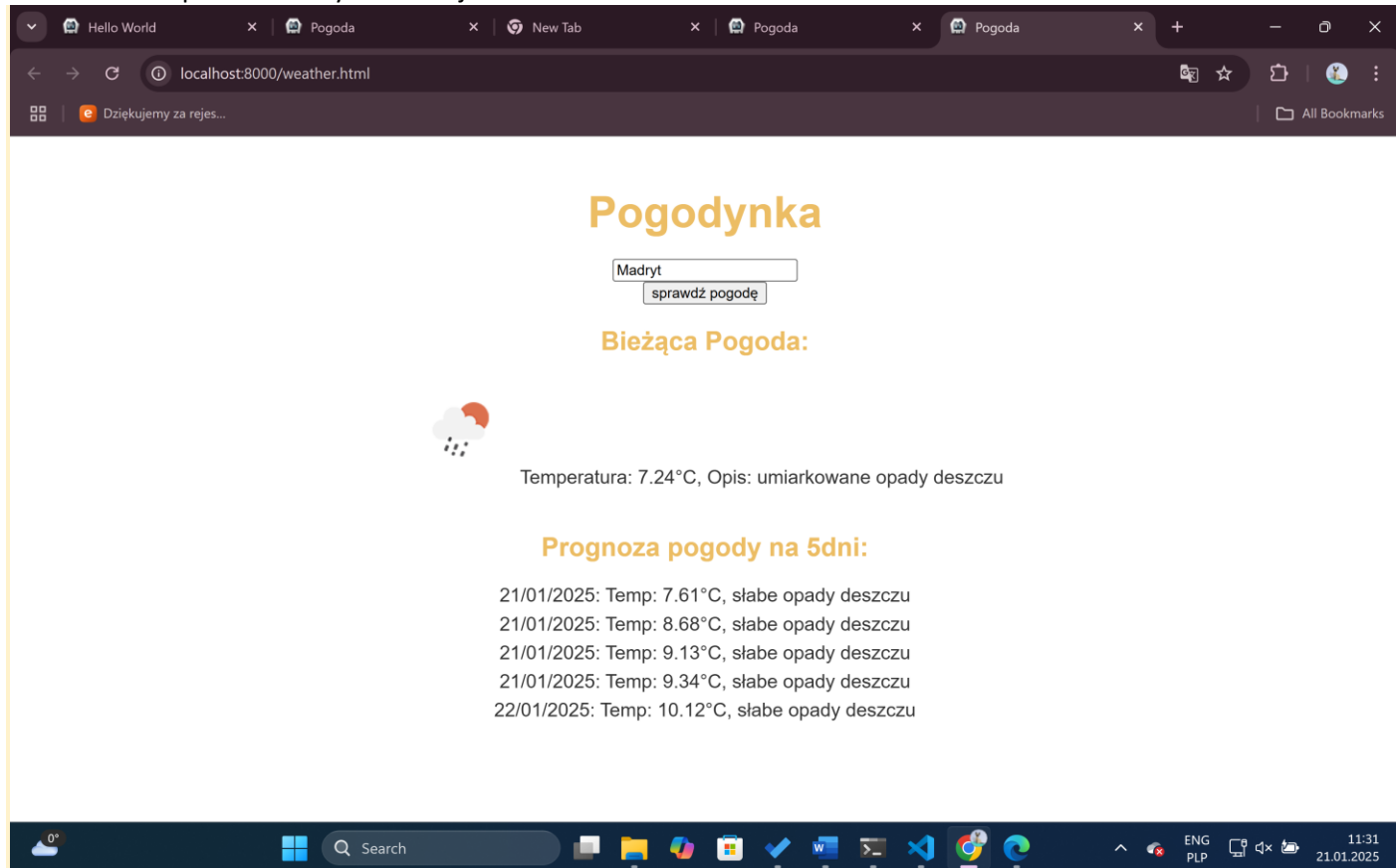
Po uruchomieniu emulacji, w przeglądarce powinna pojawić się aplikacja Pogodynki. Wprowadź nazwę miejscowości i wyszukaj prognozy pogody, przykładowo:



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:

```
c53757 > config.xml
1  <?xml version='1.0' encoding='utf-8'?>
2  <widget id="pl.edu.zut.wi.kj53757" version="1.0.0" xmlns="http://www.w3.org/ns/widgets" xmlns:cdv="http://cordova.apache.org/2011/html" >
3      <name>Judyta 53757</name>
4      <description>Weather App</description>
5      <author email="kj53757@zut.edu.pl" href="https://www.wi.zut.edu.pl">
6          Judyta Kopaniecka
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:

```

on getCurrentWeather(location) {
  nst apiKey = '71c0d0e44ebf4e273a3cc2213c575a67';
  nst url = `https://api.openweathermap.org/data/2.5/weather?q=${location}&appid=${apiKey}&units=metric&lang=pl`;

  nst xhr = new XMLHttpRequest();
  r.open("GET", url, true);
  r.onreadystatechange = function () {
    if (xhr.readyState === 4 && xhr.status === 200) {
      const data = JSON.parse(xhr.responseText);
      console.log("Bieżąca pogoda:", data);
      drawWeather(data);
    }
  }

  r.send();

on drawWeather(data) {
  cument.getElementById('weatherInfo').innerHTML = `
    
    Temperatura: ${data.main.temp}°C,
    Opis: ${data.weather[0].description}

on getForecast(location) {
  nst apiKey = '71c0d0e44ebf4e273a3cc2213c575a67';
  nst url = `https://api.openweathermap.org/data/2.5/forecast?q=${location}&appid=${apiKey}&units=metric&lang=pl`;

```

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

ANDROID

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

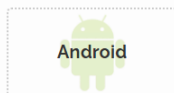
Najpierw w konsoli dodaj platformę Android:

```
> cordova platform add android
```

Następnie utwórz archiwum ZIP z całym katalogiem projektu, tj. C:\Users\...\Desktop\ai1-cordova\c53757. Załóżmy, że otrzymane archiwum to c53757.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:



Ola 12345

Android debug

Wed Jan 15 2025 16:09:24 GMT+0100 (czas środkowoeuropejski standardowy)



Download



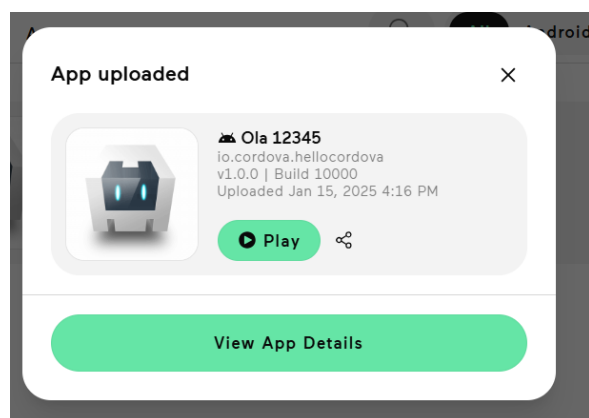
Copy URL for use on device to clipboard



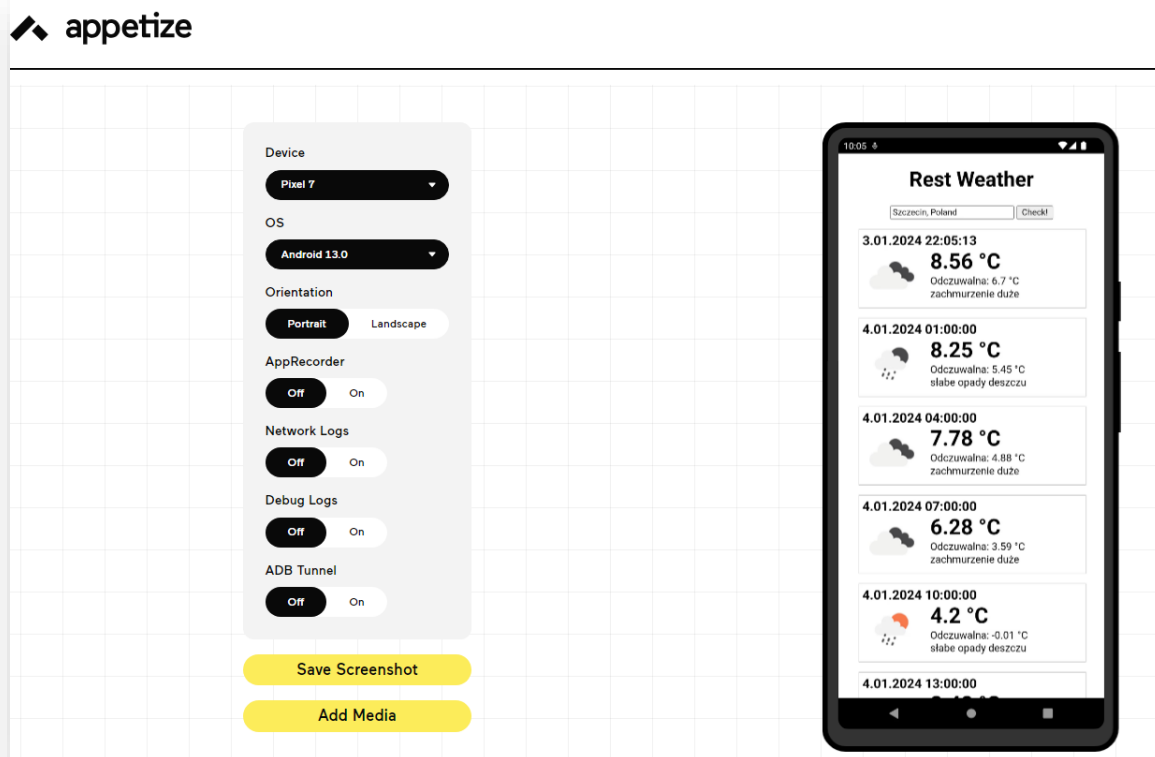
View Log

Pobierz aplikację. Będzie to plik `.apk`.

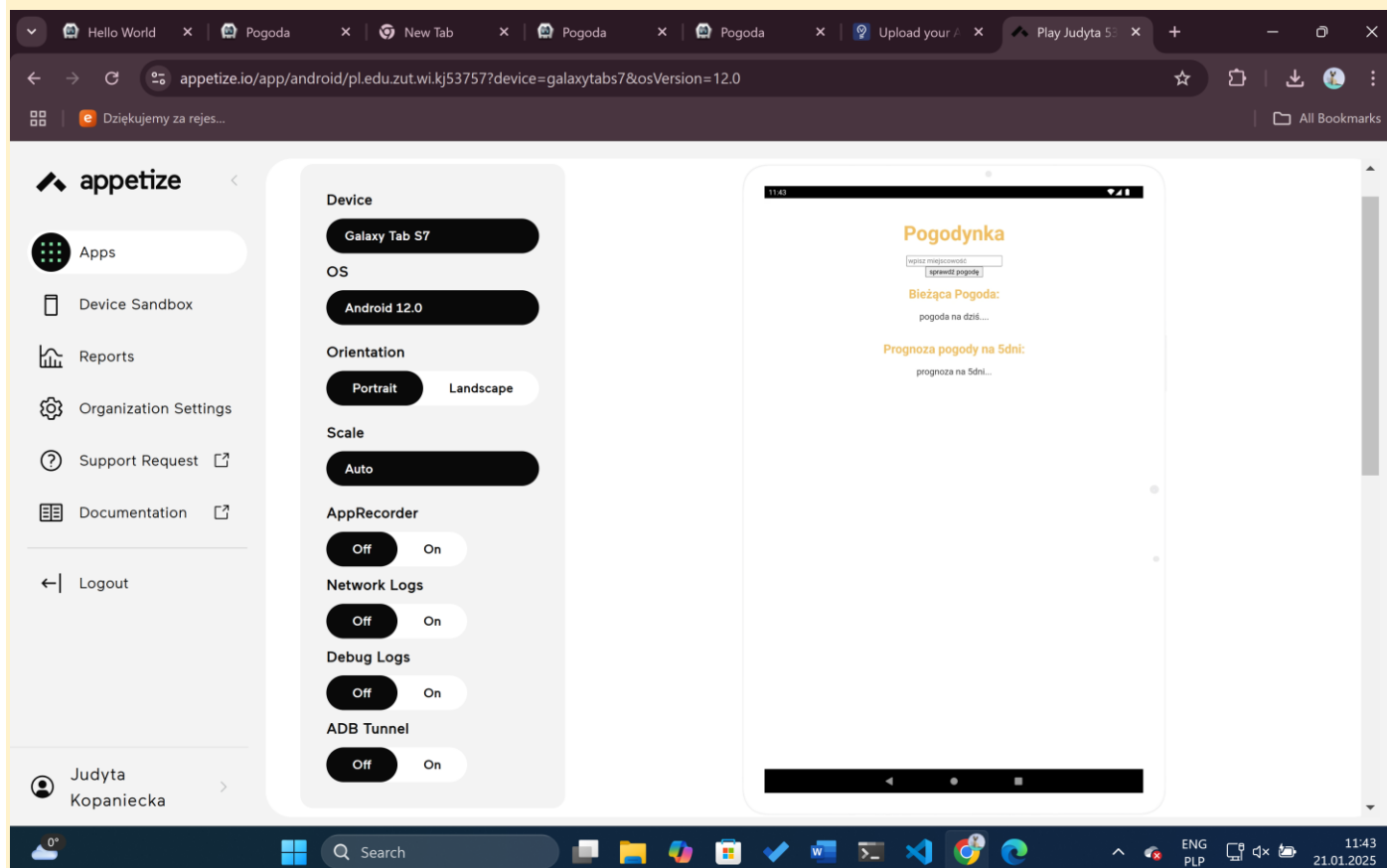
Teraz jesteśmy gotowi do przetestowania działania aplikacji w emulatorze Android. Wykorzystamy do tego celu usługę [appetize](https://appetize.io/). Wejdź na stronę <https://appetize.io/>. Utwórz darmowe konto. Przejdź do **Demo**. Przyciśnij **Upload App** i wybierz z dysku swoją aplikację. W ciągu kilku chwil otworzy się okno z informacją **App uploaded**:



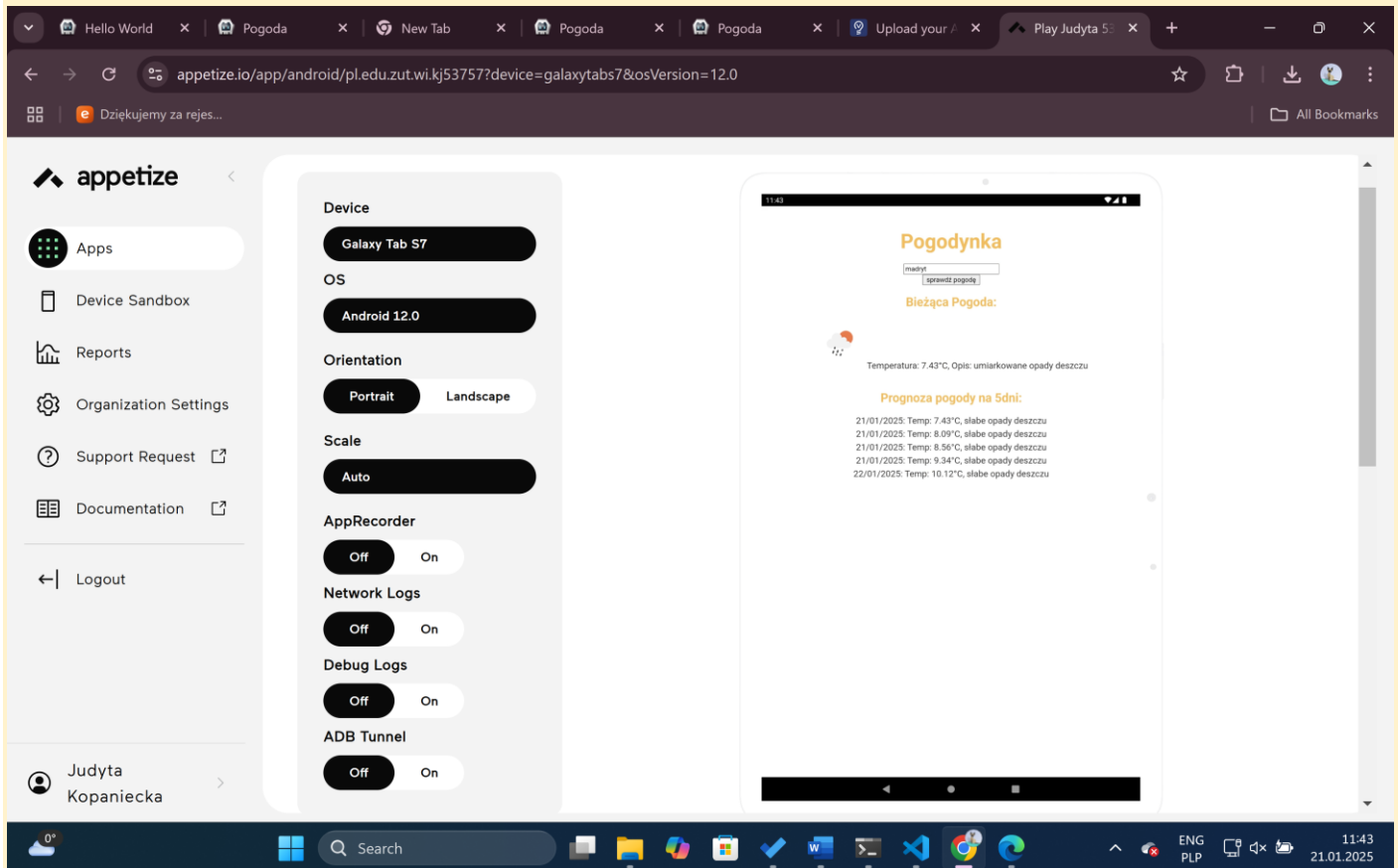
Kliknij przycisk Play. Wybierz urządzenie, system operacyjny i uruchom emulację:



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



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





Wstaw zrzut ekranu z załadowaną aplikację **appetize** (okno z informacją App Uploaded / listę aplikacji). Upewnij się, że widoczna jest Twoja aplikacja a przy niej jest Twoje imię i numer albumu:

Apps



All Apps

App Groups



 **Judyta 53757**
pl.edu.zut.wi.kj53757
v1.0.0 (2 minutes ago)

Last played 2 minutes ...

 **Play** 

Wstaw zrzut ekranu fragmentu strony **VoltBuilder** przedstawiający zbudowaną aplikację. Upewnij się, że widoczne jest Twoje imię i numer albumu:

**Judyta 53757**

Android debug

Tue Jan 21 2025 11:37:06 GMT+0100 (Central European Standard Time)



Download



Copy URL for use on device to clipboard

[View Log](#)

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"
      ]
    }
  }
}
```

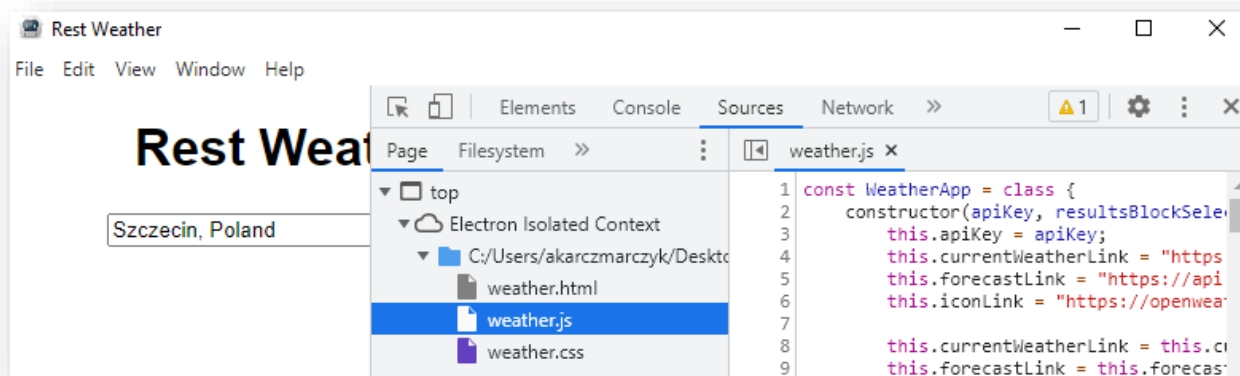
Przy budowie aplikacji, biblioteka Electron tworzyć będzie archiwum ZIP z projektem, zamiast instalatora. Dostępne postaci pakietów instalacyjnych opisane są tutaj:

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

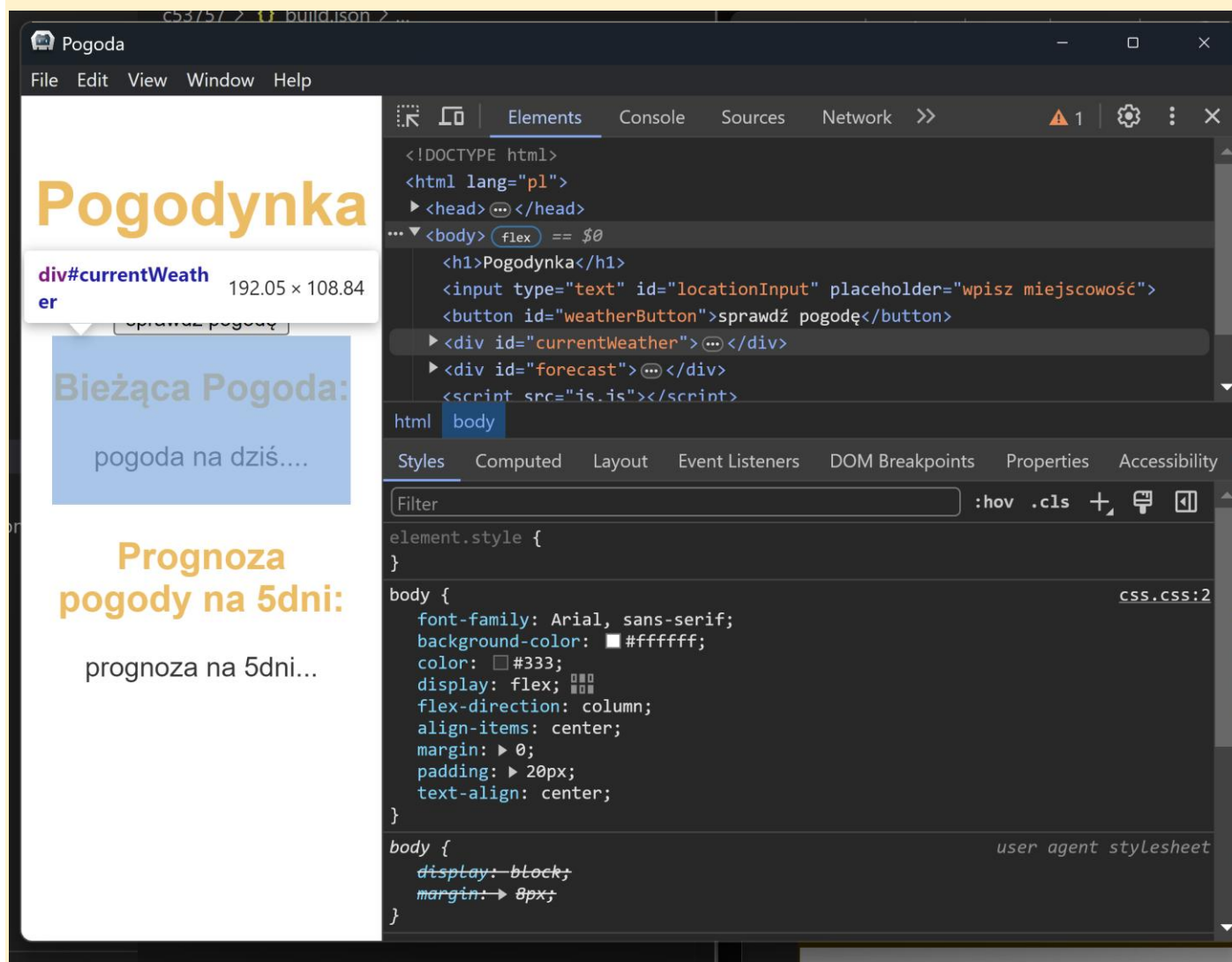
Uruchom teraz emulację aplikacji Electron:

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

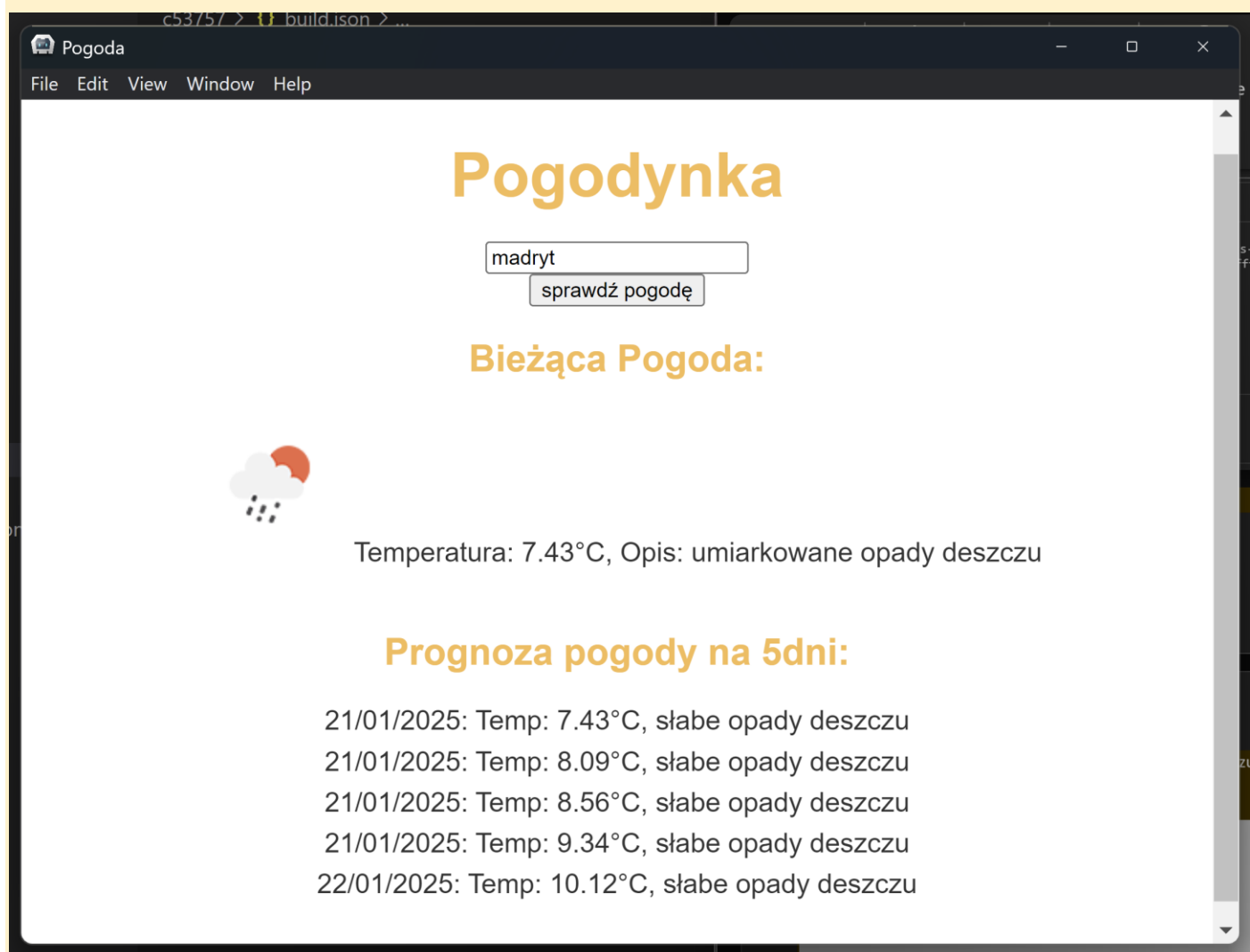
W efekcie uruchomi się aplikacja Electron:



Wstaw zrzut ekranu uruchomionej aplikacji Electron. Upewnij się że widoczne będzie całe okno wraz z ikoną Apache Cordova :



Zamknij narzędzia deweloperskie (panel z prawej strony) i wyszukaj prognozę pogody. Upewnij się że widoczne będzie całe okno wraz z ikoną Apache Cordova. Wstaw zrzut ekranu:

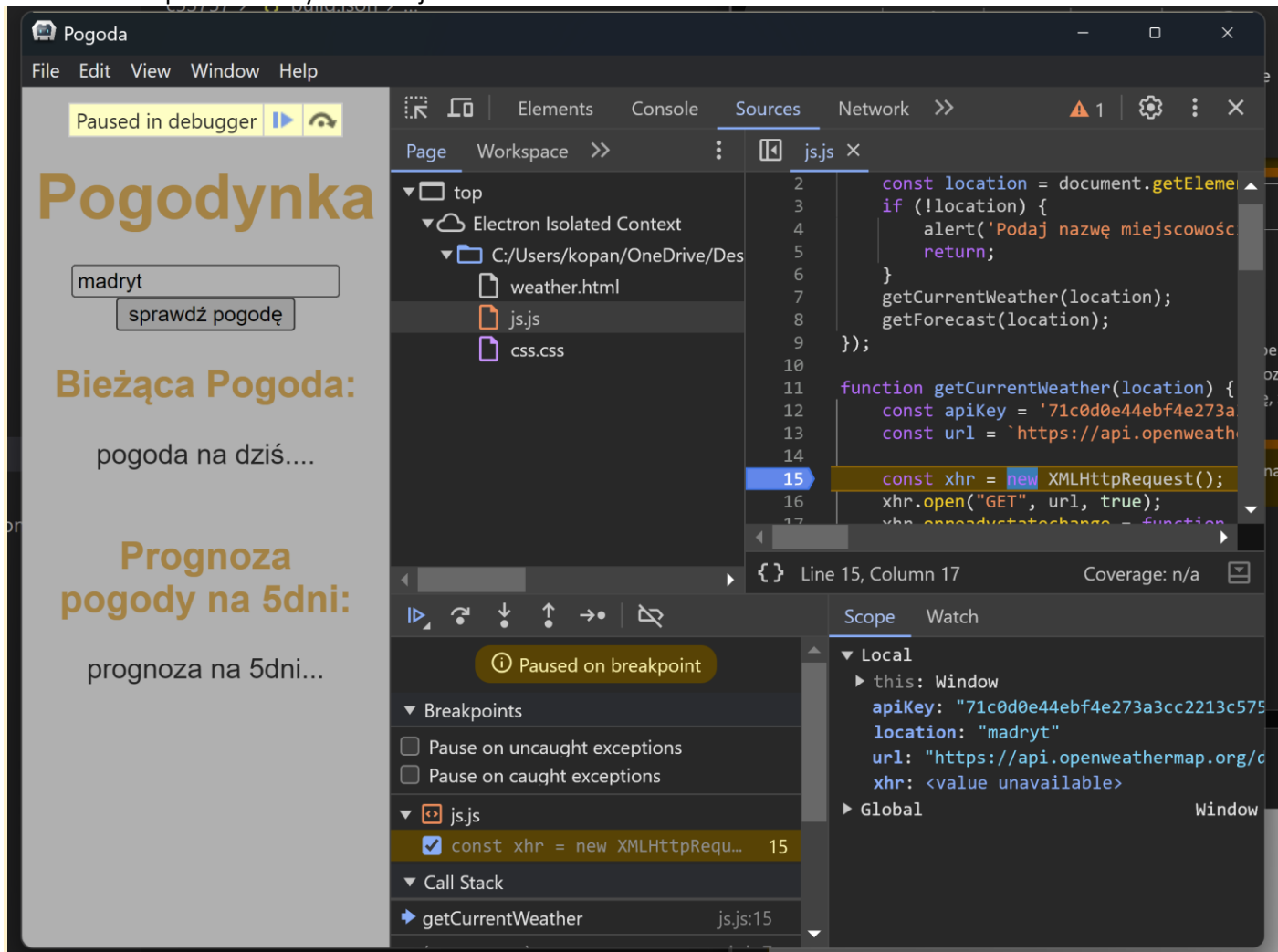


Punkty:	0	1
---------	---	---

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 / żółto)
- 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.

Uwaga! W przypadku niektórych systemów może wystąpić błąd:

Cannot create symbolic link : A required privilege is not held by the client.

Szczegółowy opis rozwiązania znajduje się tu:

<https://stackoverflow.com/questions/78560953/i-am-get-this-error-while-build-my-electron-vite-project>

W skrócie:

- pobierz plik <https://github.com/electron-userland/electron-builder-binaries/releases/download/winCodeSign-2.6.0/winCodeSign-2.6.0.7z>
- rozpakuj go do katalogu zgodnego z komunikatem błędu, np. `C:\Users\artur\AppData\Local\electron-builder\Cache\winCodeSign\winCodeSign-2.6.0`
- uruchom ponownie budowanie aplikacji

Jak nic nie działa (absolutnie nie polecane rozwiązanie):

- uruchom terminal jako administrator i spróbuj ponownie (nie da się w sieci ZUT)

Wstaw zrzut ekranu kodu pliku `settings.json`:

```

c53757 > res > electron > {} settings.json > ...

1      {
2      |      "browserWindow": {
3      |          "width": 1024,
4      |          "height": 600,
5      |          "autoHideMenuBar": true
6      |      }
7      }
8

```

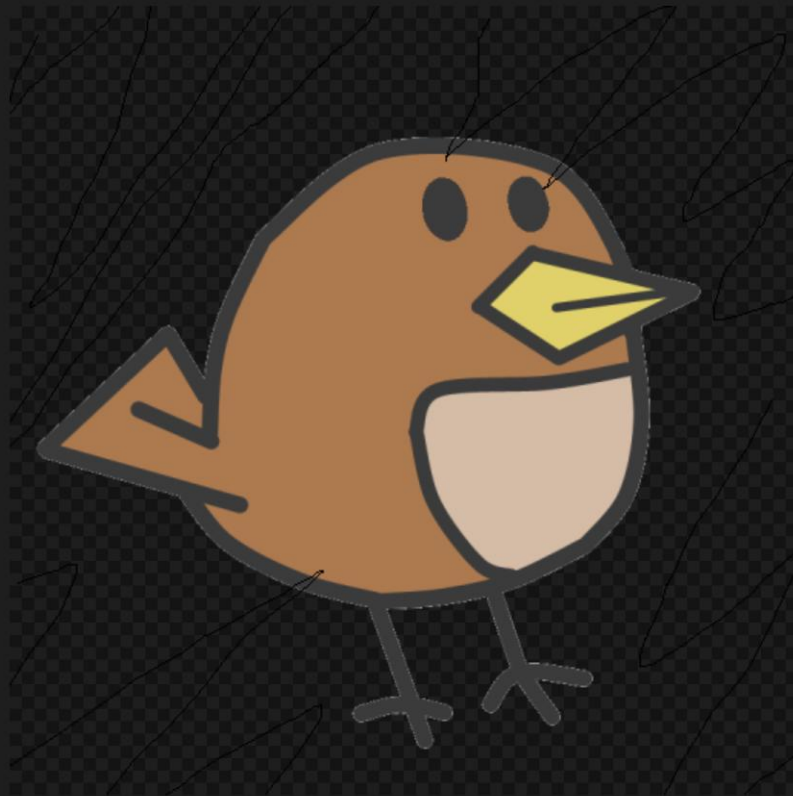
Wstaw zrzut ekranu kodu pliku `config.xml`:

```

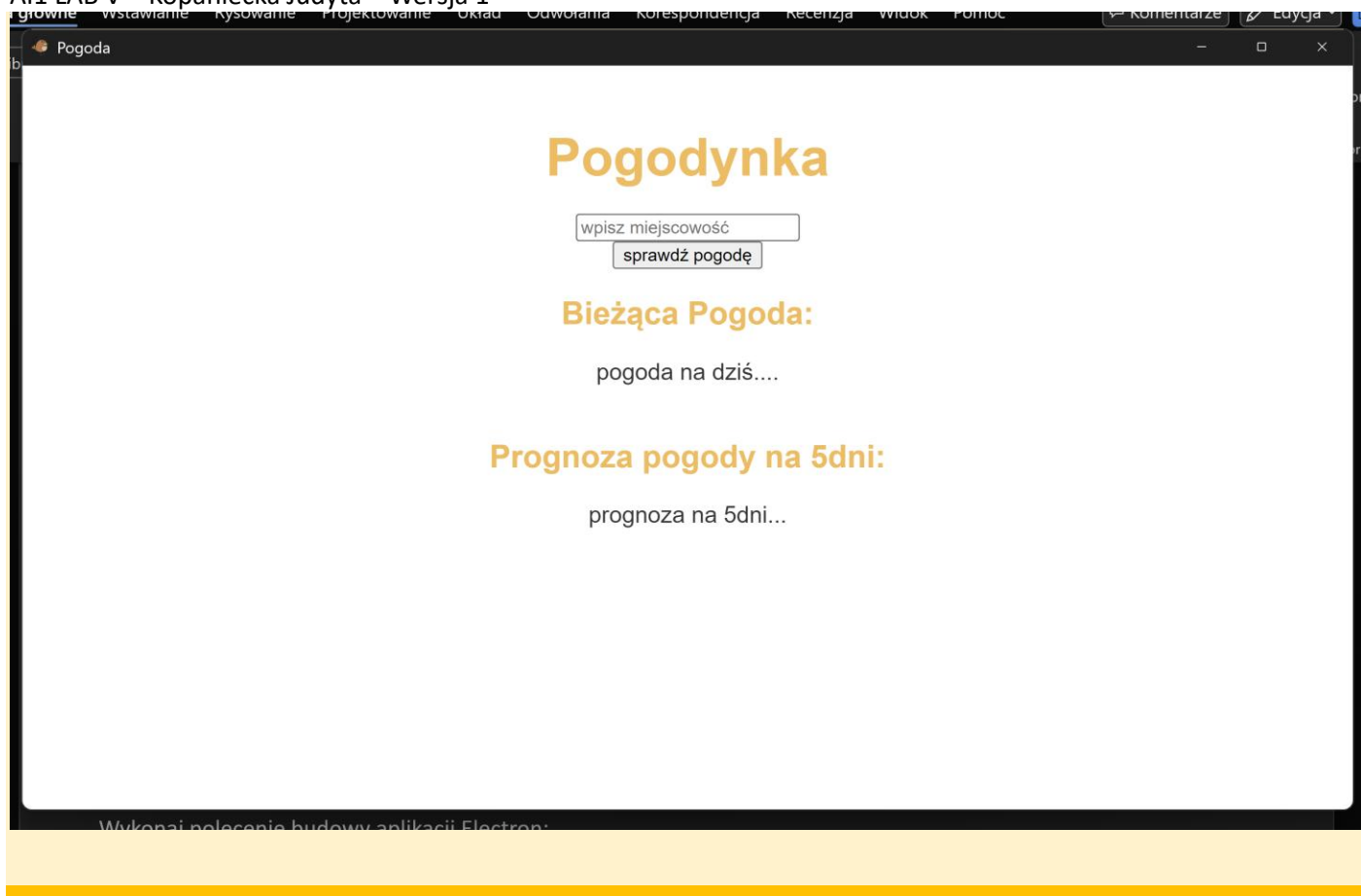
c53757 > config.xml
1  <?xml version='1.0' encoding='utf-8'?>
2  <widget id="pl.edu.zut.wi.kj53757" version="1.0.0" xmlns="http://www.w3.org/ns/widgets" xmlns:cdv="http://cordova.apache.org/2011/html" >
3      <name>Judyta 53757</name>
4      <description>Weather App</description>
5      <author email="kj53757@zut.edu.pl" href="https://www.wi.zut.edu.pl">
6          Judyta Kopaniecka
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/icon.png" />
13         <preference name="ElectronSettingsFilePath" value="res/electron/settings.json" />
14     </platform>
15
16 </widget>
17

```

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. **WAŻNE!!!** Użytkownicy MacOS – zrób zrzut ekranu w taki sposób, żeby ikonka uruchomionej aplikacji na pewno była widoczna – jeśli jest poza oknem – obejmij screenshotem cały ekran i zaznacz ikonkę.



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

BUDOWA APLIKACJI

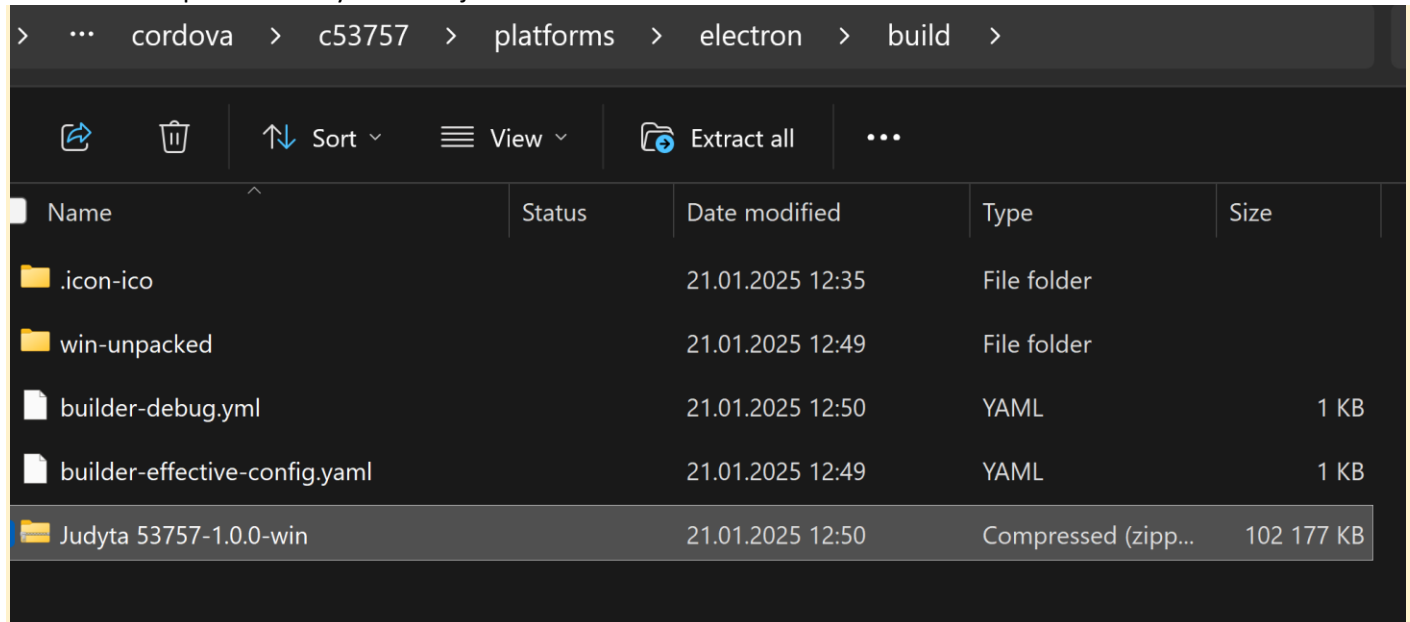
Wykonaj polecenie budowy aplikacji Electron:

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

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

Skopiuj archiwum do katalogu `Pobrane`. Rozpakuj archiwum do katalogu `Judyta 53757-1.0.0-win`, wejdź do środka rozpakowanego katalogu i uruchom plik EXE.






Wstaw zrzut ekranu zawartości katalogu `platforms\electron\build`. Zaznacz na zrzucie ekranu plik `Judyta 53757-1.0.0-win.zip`





















> ... cordova > c53757 > platforms > electron > build >					
Sort View Extract all ...					
Name	Status	Date modified	Type	Size	
.icon-ico		21.01.2025 12:35	File folder		
win-unpacked		21.01.2025 12:49	File folder		
builder-debug.yml		21.01.2025 12:50	YAML	1 KB	
builder-effective-config.yml		21.01.2025 12:49	YAML	1 KB	
Judyta 53757-1.0.0-win		21.01.2025 12:50	Compressed (zipp...	102 177 KB	

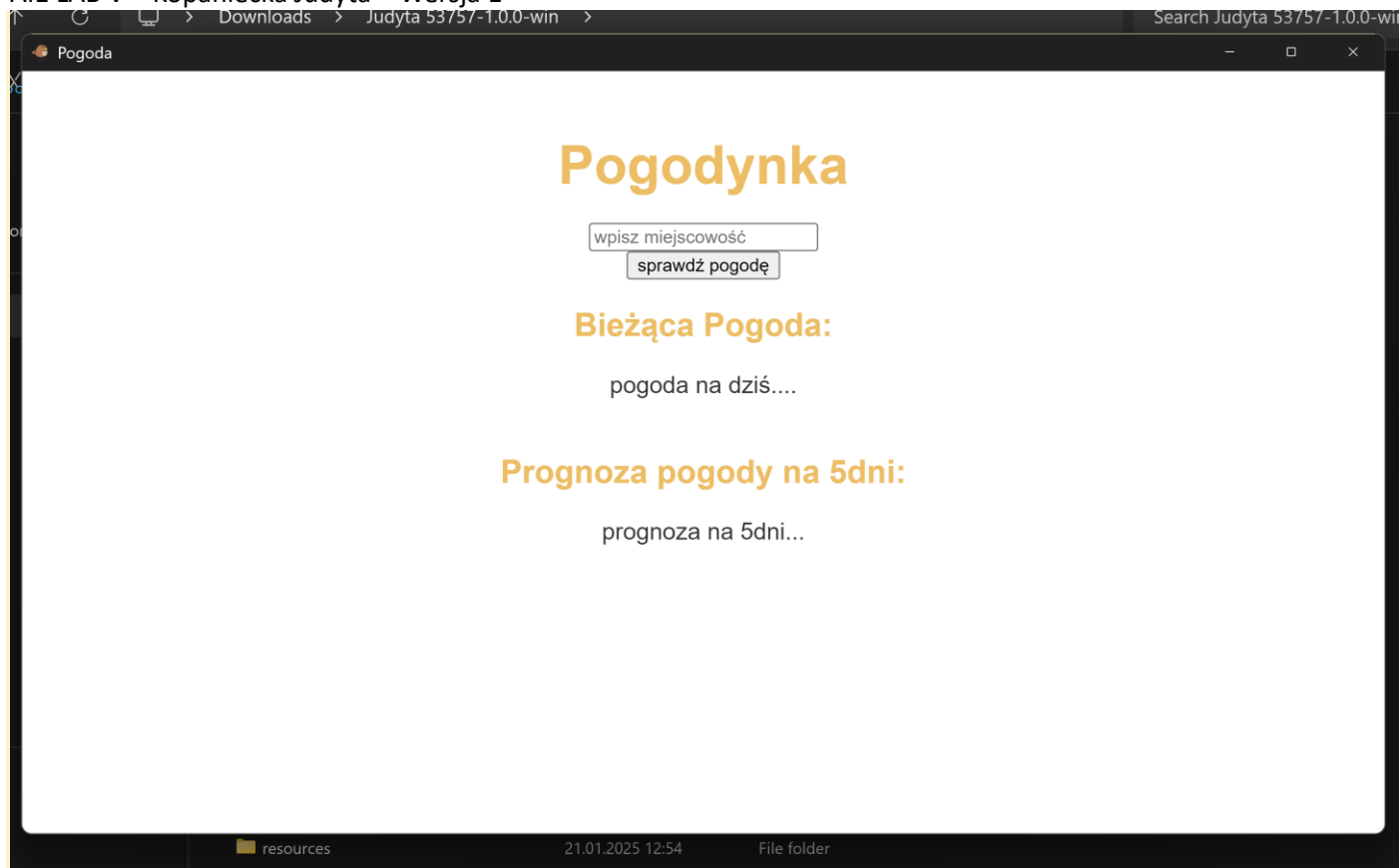
Wstaw zrzut ekranu zawartości rozpakowanego archiwum ZIP z aplikacją. Upewnij się, że na zrzucie ekranu jest ścieżka do katalogu. Zaznacz na zrzucie ekranu plik **Judyta 53757.exe**. Upewnij się, że przy pliku widoczna jest Twoja ikona.

Downloads > Judyta 53757-1.0.0-win >




 Sort ▾
  View ▾
 ...

<input type="checkbox"/> Name	Date modified	Type	Size
▼ Today			
<input checked="" type="checkbox"/>  Judyta 53757	21.01.2025 12:49	Application	172 171 KB
 chrome_100_percent.pak	21.01.2025 12:49	PAK File	151 KB
 chrome_200_percent.pak	21.01.2025 12:49	PAK File	230 KB
 d3dcompiler_47.dll	21.01.2025 12:49	Application extens...	4 802 KB
 ffmpeg.dll	21.01.2025 12:49	Application extens...	2 799 KB
 icudtl.dat	21.01.2025 12:49	DAT File	10 467 KB
 libEGL.dll	21.01.2025 12:49	Application extens...	468 KB
 libGLESv2.dll	21.01.2025 12:49	Application extens...	7 513 KB
 LICENSE.electron	21.01.2025 12:49	Text Document	2 KB
 LICENSES.chromium	21.01.2025 12:49	Microsoft Edge H...	8 960 KB
 resources.pak	21.01.2025 12:49	PAK File	5 158 KB
 snapshot_blob.bin	21.01.2025 12:49	BIN File	300 KB
 v8_context_snapshot.bin	21.01.2025 12:49	BIN File	664 KB
 vk_swiftshader.dll	21.01.2025 12:49	Application extens...	5 188 KB
 vk_swiftshader_icd.json	21.01.2025 12:49	JSON	1 KB
 vulkan-1.dll	21.01.2025 12:49	Application extens...	932 KB
 locales	21.01.2025 12:54	File folder	
 resources	21.01.2025 12:54	File folder	

Wstaw zrzut ekranu uruchomionej aplikacji przed sprawdzeniem pogody:



Wstaw zrzut ekranu uruchomionej aplikacji po sprawdzeniu prognozy pogody:



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:

https://github.com/jkopanieckaa/aplikacje_internetowe/tree/main/cordova

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

PODSUMOWANIE

W kilku zdaniach podsumuj zdobyte podczas tego laboratorium umiejętności.

Dowiedziałam się jak można budować aplikacje

...podsumowanie...

Zweryfikuj kompletność sprawozdania. Utwórz PDF i wyślij w terminie.