|  |  |  |
| --- | --- | --- |
| **AI1** | Kopaniecka Judyta  Album 53757  Grupa 330 | **Wersja 1** |
|  |
| **LAB V** |

Cordova

# Spis treści

[Spis treści 1](#_Toc188348498)

[Cel zajęć 1](#_Toc188348499)

[Uwaga 1](#_Toc188348500)

[Instalacja Cordova 2](#_Toc188348501)

[Utworzenie projektu Cordova i pierwsze uruchomienie 2](#_Toc188348502)

[Podmiana projektu domyślnego na Pogodynkę 5](#_Toc188348503)

[Android 8](#_Toc188348504)

[Electron i aplikacja na Windows 13](#_Toc188348505)

[Debuggowanie aplikacji Electron 15](#_Toc188348506)

[Dostosowanie aplikacji Electron 16](#_Toc188348507)

[Budowa aplikacji 20](#_Toc188348508)

[Commit projektu do GIT 24](#_Toc188348509)

[Podsumowanie 24](#_Toc188348510)

# 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:

Obraz zawierający tekst, zrzut ekranu, oprogramowanie, Oprogramowanie multimedialne

Opis wygenerowany automatycznie

Wstaw zrzut ekranu emulacji w przeglądarce domyślnego projektu:

Obraz zawierający tekst, zrzut ekranu, oprogramowanie, Oprogramowanie multimedialne

Opis wygenerowany automatycznie

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

Obraz zawierający tekst, zrzut ekranu

Opis wygenerowany automatycznie

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:

> cordova emulate browser

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

Obraz zawierający tekst, zrzut ekranu, oprogramowanie, Ikona komputerowa

Opis wygenerowany automatycznie

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

Obraz zawierający tekst, zrzut ekranu, oprogramowanie, komputer

Opis wygenerowany automatycznie

Wstaw zrzut ekranu emulacji pogodynki po wyszukaniu prognoz:

Obraz zawierający tekst, elektronika, zrzut ekranu, oprogramowanie

Opis wygenerowany automatycznie

Wstaw zrzut ekranu pliku config.xml:

Obraz zawierający tekst, zrzut ekranu

Opis wygenerowany automatycznie

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

Obraz zawierający tekst, zrzut ekranu, Czcionka

Opis wygenerowany automatycznie

|  |  |  |
| --- | --- | --- |
| 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:

Obraz zawierający tekst, zrzut ekranu, diagram

Opis wygenerowany automatycznie

Po kilku minutach aplikacja będzie gotowa do pobrania:

Obraz zawierający tekst, zrzut ekranu, Czcionka, design

Opis wygenerowany automatycznie

Pobierz aplikację. Będzie to plik .apk.

Teraz jesteśmy gotowi do przetestowania działania aplikacji w emulatorze Android. Wykorzystamy do tego celu usługę appetize. Wejdź na strone <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**:

Obraz zawierający tekst, zrzut ekranu, gadżet, multimedia

Opis wygenerowany automatycznie

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

Obraz zawierający tekst, zrzut ekranu, numer

Opis wygenerowany automatycznie

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

Obraz zawierający tekst, zrzut ekranu, oprogramowanie, Oprogramowanie multimedialne

Opis wygenerowany automatycznie

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

Obraz zawierający tekst, zrzut ekranu, oprogramowanie, Oprogramowanie multimedialne

Opis wygenerowany automatycznie

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:

Obraz zawierający tekst, zrzut ekranu, Marka, logo

Opis wygenerowany automatycznie

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

Obraz zawierający tekst, zrzut ekranu, Czcionka, design

Opis wygenerowany automatycznie

|  |  |  |
| --- | --- | --- |
| 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:

Obraz zawierający tekst, zrzut ekranu, Czcionka, numer

Opis wygenerowany automatycznie

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

Obraz zawierający tekst, elektronika, zrzut ekranu, oprogramowanie

Opis wygenerowany automatycznie

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

Obraz zawierający tekst, elektronika, zrzut ekranu, oprogramowanie

Opis wygenerowany automatycznie

|  |  |  |
| --- | --- | --- |
| 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 breakpoincie:

Obraz zawierający tekst, oprogramowanie, zrzut ekranu, komputer

Opis wygenerowany automatycznie

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:

Obraz zawierający tekst, zrzut ekranu, Czcionka, numer

Opis wygenerowany automatycznie

Wstaw zrzut ekranu kodu pliku config.xml:

Obraz zawierający tekst, zrzut ekranu, wyświetlacz, oprogramowanie

Opis wygenerowany automatycznie

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

Obraz zawierający rysowanie, clipart, kreskówka, ptak

Opis wygenerowany automatycznie

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.

Obraz zawierający tekst, zrzut ekranu, oprogramowanie, komputer

Opis wygenerowany automatycznie

|  |  |  |
| --- | --- | --- |
| 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

Obraz zawierający tekst, zrzut ekranu, oprogramowanie, Oprogramowanie multimedialne

Opis wygenerowany automatycznie

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.

Obraz zawierający tekst, zrzut ekranu, oprogramowanie, Oprogramowanie multimedialne

Opis wygenerowany automatycznie

Wstaw zrzut ekranu uruchomionej aplikacji przed sprawdzeniem pogody:

Obraz zawierający tekst, elektronika, zrzut ekranu, oprogramowanie

Opis wygenerowany automatycznie

Wstaw zrzut ekranu uruchomionej aplikacji po sprawdzeniu prognozy pogody:

Obraz zawierający tekst, zrzut ekranu, oprogramowanie, Strona internetowa

Opis wygenerowany automatycznie

|  |  |  |
| --- | --- | --- |
| 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.