

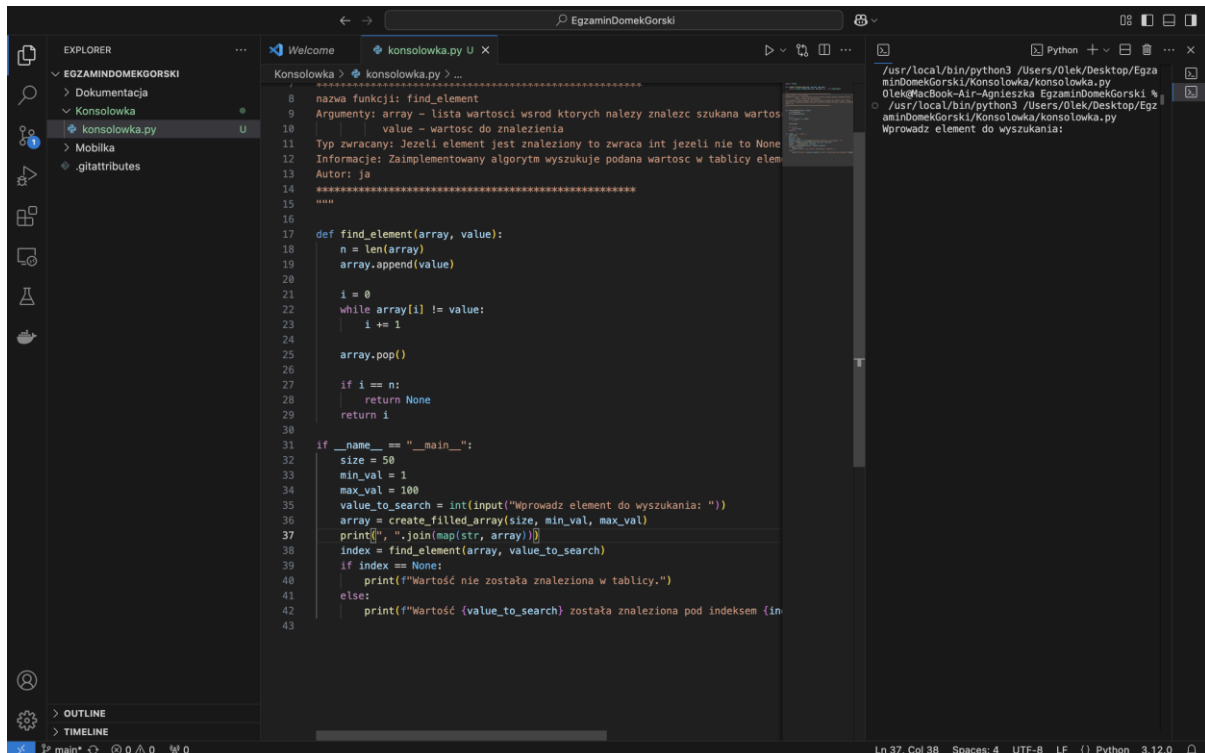
System Operacyjny: MacOS Big Sur Wersja 11.4

Srodowiska programistyczne: Android Studio, Visual Studio Code

Jezyki programowania: Python, Kotlin

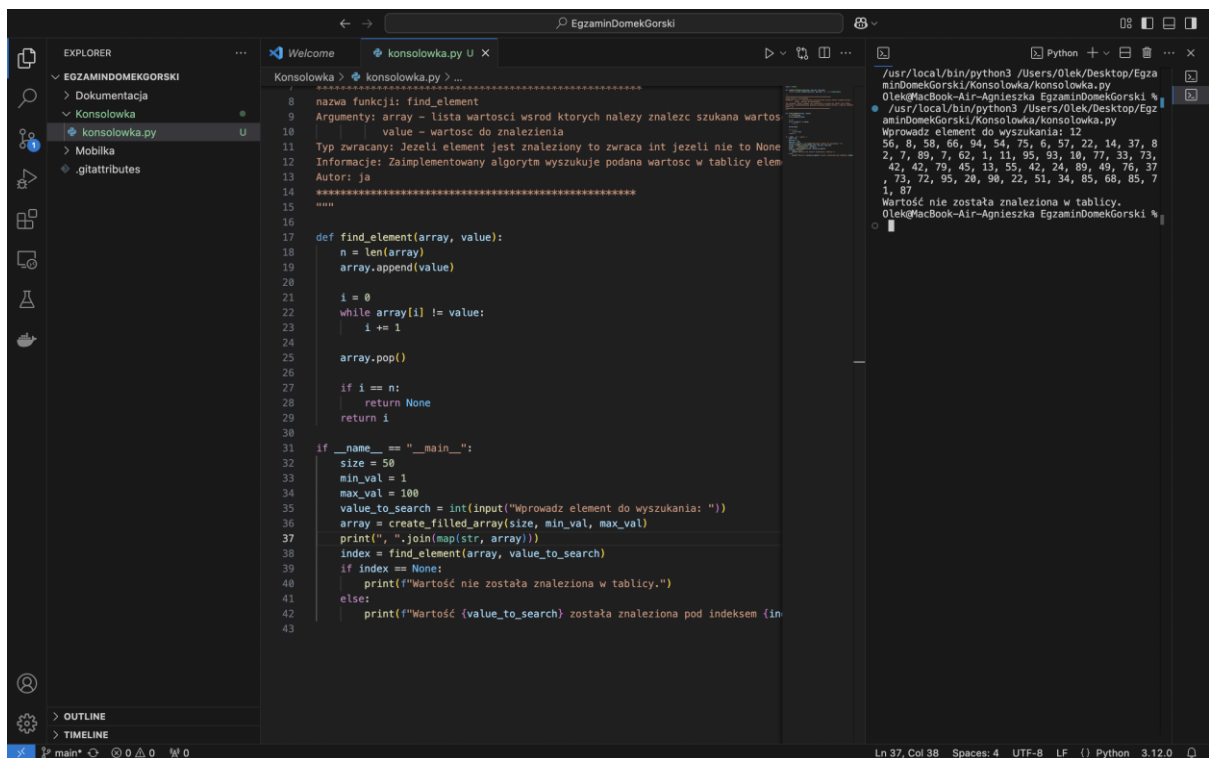
Emulator systemu mobilnego: Pixel 7 Pro API 31

Aplikacja Konsolowa:



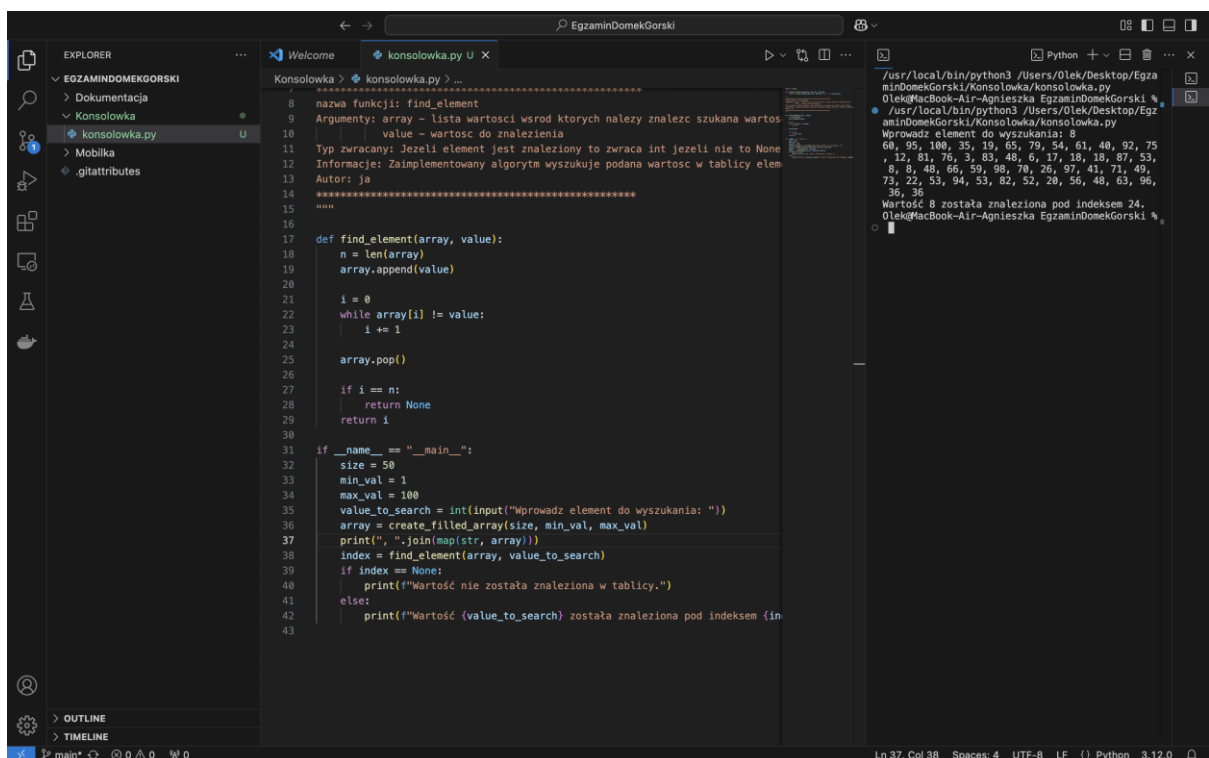
```
8 nazwa funkcji: find_element
9 Argumenty: array - lista wartosci wsrod ktorych nalezy znalezc szukana wartosc
10 value - wartosc do znalezienia
11 Typ zwracany: Jezeli element jest znaleziony to zwraca int jezeli nie to None
12 Informacje: Zaimplementowany algorytm wyszukuje podana wartosc w tablicy elem
13 Autor: ja
14 *****
15
16
17 def find_element(array, value):
18     n = len(array)
19     array.append(value)
20
21     i = 0
22     while array[i] != value:
23         i += 1
24
25     array.pop()
26
27     if i == n:
28         return None
29     return i
30
31 if __name__ == "__main__":
32     size = 50
33     min_val = 1
34     max_val = 100
35     value_to_search = int(input("Wprowadz element do wyszukania: "))
36     array = create_filled_array(size, min_val, max_val)
37     print(" ".join(map(str, array)))
38     index = find_element(array, value_to_search)
39     if index == None:
40         print("Wartosc nie zostala znaleziona w tablicy.")
41     else:
42         print("Wartosc {value_to_search} zostala znaleziona pod indeksem {in
43
```

Aplikacja Po uruchomieniu



```
8 nazwa funkcji: find_element
9 Argumenty: array - lista wartosci wsrod ktorch nalezy znalezc szukana wartosc
10 | value - wartosc do znalezienia
11 Typ zwracany: Jezeli element jest znaleziony to zwraca int jezeli nie to None
12 Informacje: Zaimplementowany algorytm wyszukuje podana wartosc w tablicy elem
13 Autor: ja
14 *****
15
16
17 def find_element(array, value):
18     n = len(array)
19     array.append(value)
20
21     i = 0
22     while array[i] != value:
23         i += 1
24
25     array.pop()
26
27     if i == n:
28         return None
29     return i
30
31 if __name__ == "__main__":
32     size = 50
33     min_val = 1
34     max_val = 100
35     value_to_search = int(input("Wprowadz element do wyszukania: "))
36     array = create_filled_array(size, min_val, max_val)
37     print(" ".join(map(str, array)))
38     index = find_element(array, value_to_search)
39     if index == None:
40         print(f"Wartosc nie zostala znaleziona w tablicy.")
41     else:
42         print(f"Wartosc {value_to_search} zostala znaleziona pod indeksem {in
43
```

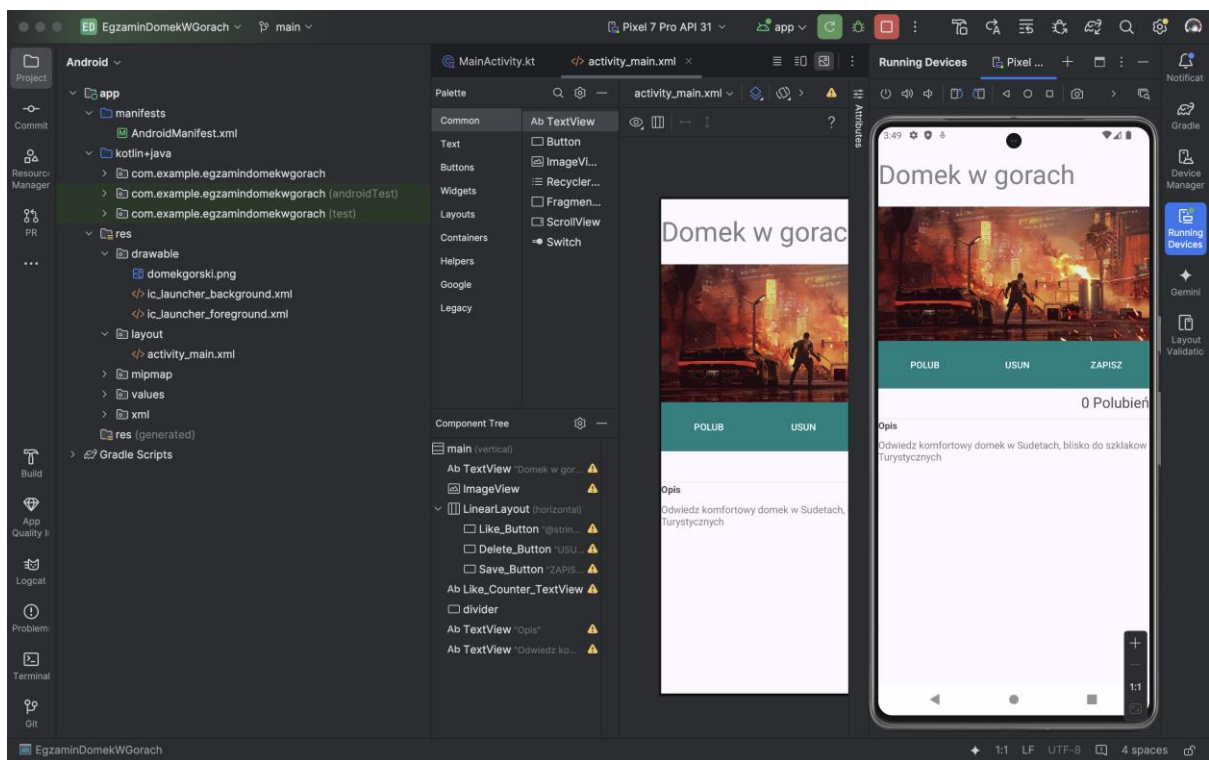
Zrzut ekranu przedstawia Aplikacje Gdy wyszukany element nie znajduje sie na liscie



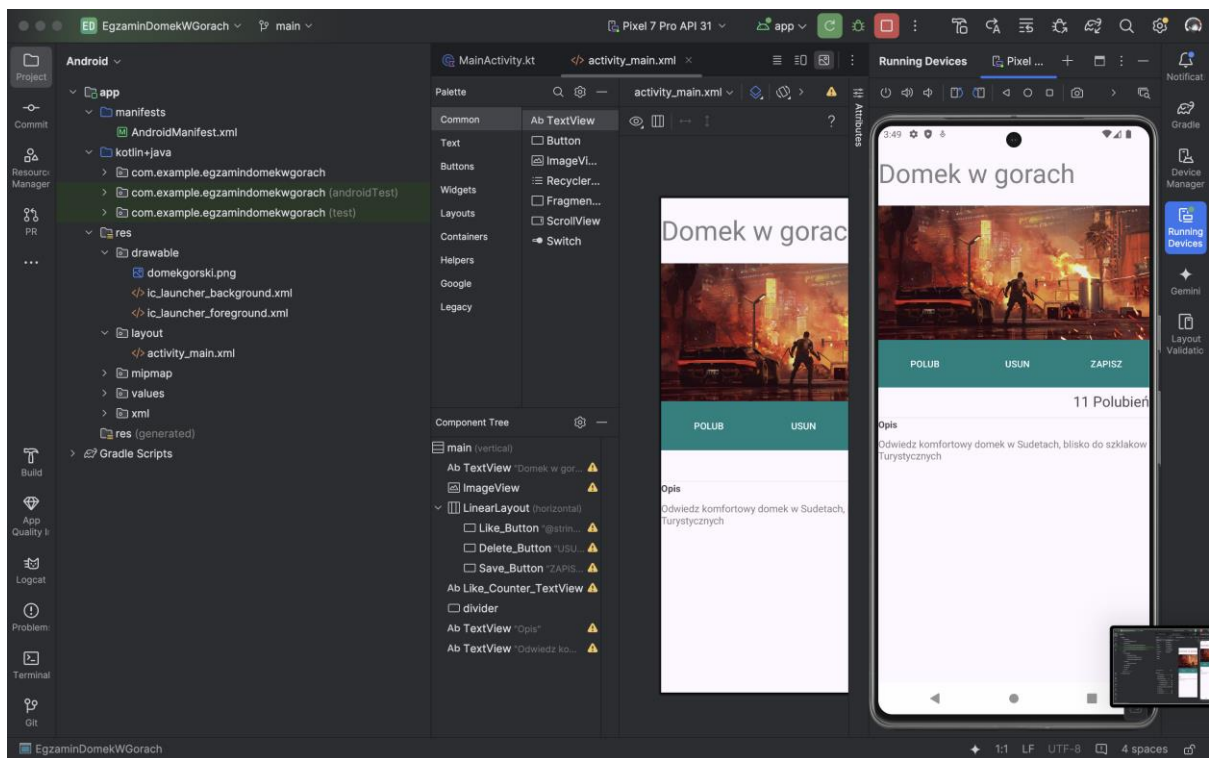
```
8 nazwa funkcji: find_element
9 Argumenty: array - lista wartosci wsrod ktorch nalezy znalezc szukana wartosc
10 | value - wartosc do znalezienia
11 Typ zwracany: Jezeli element jest znaleziony to zwraca int jezeli nie to None
12 Informacje: Zaimplementowany algorytm wyszukuje podana wartosc w tablicy elem
13 Autor: ja
14 *****
15
16
17 def find_element(array, value):
18     n = len(array)
19     array.append(value)
20
21     i = 0
22     while array[i] != value:
23         i += 1
24
25     array.pop()
26
27     if i == n:
28         return None
29     return i
30
31 if __name__ == "__main__":
32     size = 50
33     min_val = 1
34     max_val = 100
35     value_to_search = int(input("Wprowadz element do wyszukania: "))
36     array = create_filled_array(size, min_val, max_val)
37     print(" ".join(map(str, array)))
38     index = find_element(array, value_to_search)
39     if index == None:
40         print(f"Wartosc nie zostala znaleziona w tablicy.")
41     else:
42         print(f"Wartosc {value_to_search} zostala znaleziona pod indeksem {in
43
```

Zrzut ekranu przedstawia Aplikacje Gdy wyszukany element zostaje odnaleziony w liscie

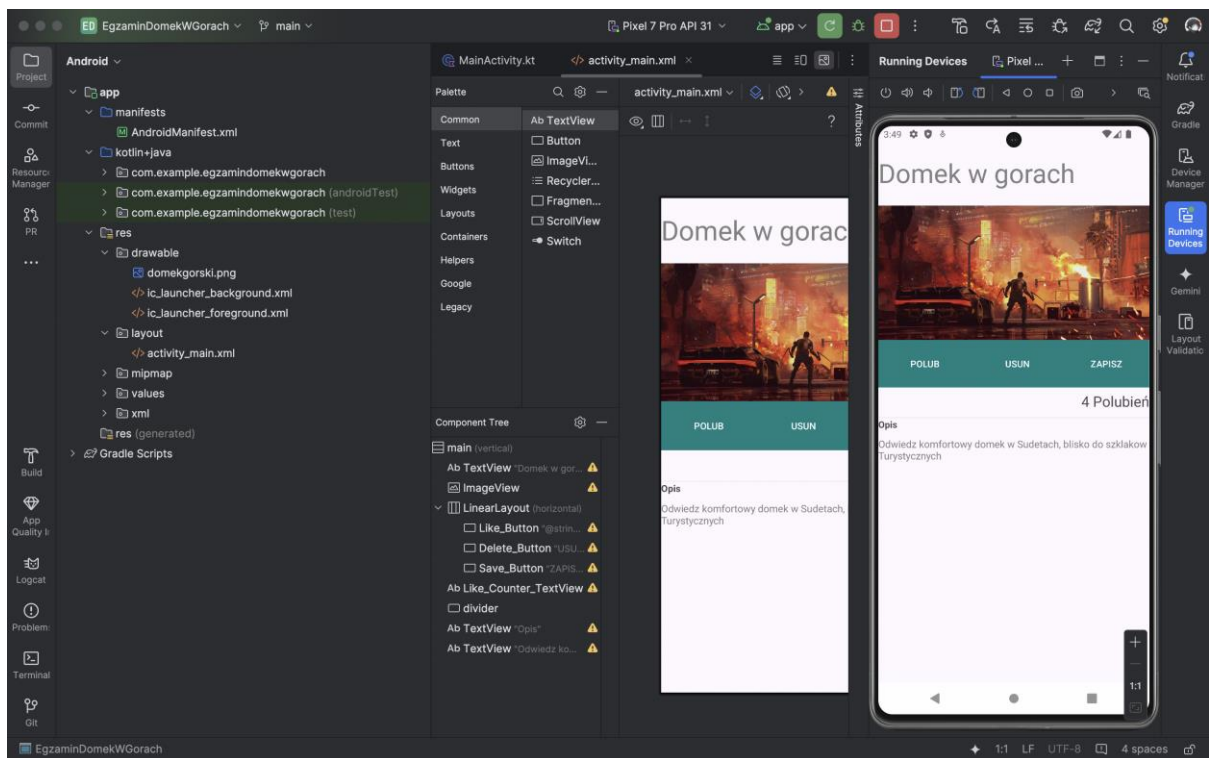
## Aplikacja Mobilna:



## Aplikacja Po Uruchomieniu



Przycisk "POLUB" został wcisnięty 11 razy



Przycisk "USUN" został wcisnięty 7 razy