



# Politechnika Świętokrzyska Kielce University of Technology

<b>Politechnika Świętokrzyska</b>		
Bezpieczeństwo aplikacji mobilnych, projekt		
<b>Kierunek:</b>	Informatyka	<b>Skład zespołu:</b>
<b>Specjalizacja:</b>	Cyberbezpieczeństwo	★ Michał Rusak ★ Bartłomiej Czech ★ Dawid Bodzon
Studia stacjonarne, II stopnia		
<b>Grupa</b>	23B, 24A	
<b>Semestr</b>	2	
<b>Link do repozytorium:</b>	<a href="https://github.com/michalrusak/BAM_Projekt">github.com/michalrusak/BAM_Projekt</a>	

## **Spis treści**

1. Wprowadzenie.....	3
2. Wykonanie projektu.....	3
1. Architektura aplikacji.....	3
2. Szyfrowanie danych lokalnych.....	3
3. Bezpieczna komunikacja z serwerem.....	4
4. Zarządzanie uprawnieniami i bezpieczeństwo aplikacji.....	4
5. Mechanizmy zdalnego zarządzania danymi.....	4
6. Testy manualne aplikacji.....	5
7. Omówienie najważniejszych sekcji kodu.....	24
8. Testowanie bezpieczeństwa.....	34
3. Wnioski.....	47

# 1. Wprowadzenie

Projekt "Aplikacja do przechowywania wrażliwych notatek z zabezpieczeniem przed atakiem Man-in-the-Middle" został zrealizowany w celu stworzenia narzędzia umożliwiającego użytkownikom bezpieczne przechowywanie notatek na urządzeniach mobilnych. Głównym celem projektu było zapewnienie najwyższego poziomu bezpieczeństwa zarówno podczas przechowywania danych lokalnie na urządzeniu, jak i w trakcie ich przesyłania między aplikacją a serwerem.

Aplikacja została opracowana z wykorzystaniem technologii React Native na potrzeby frontendu, NestJS dla backendu oraz bazy danych MongoDB. Implementacja obejmowała mechanizmy takie jak lokalne szyfrowanie notatek, zabezpieczona komunikacja HTTPS, wykrywanie zmian certyfikatów (Certificate Pinning), a także funkcje zwiększające ochronę danych, takie jak automatyczne blokowanie aplikacji czy mechanizm zdalnego usuwania danych.

# 2. Wykonanie projektu

Realizacja projektu obejmowała kilka kluczowych etapów, z których każdy wymagał wdrożenia odpowiednich technologii oraz metod zapewniających bezpieczeństwo danych. Poniżej szczegółowo opisano każdy z etapów prac:

## 1. Architektura aplikacji

W pierwszej kolejności zaprojektowano architekturę aplikacji, obejmującą zarówno część frontendową, jak i backendową.

- **Frontend** został opracowany w technologii React Native.
- **Backend** został zbudowany z użyciem frameworka NestJS.
- Dane przechowywane są w bazie danych MongoDB.

## 2. Szyfrowanie danych lokalnych

Jednym z głównych założeń projektu było zabezpieczenie danych lokalnych.

- Wykorzystano bibliotekę **react-native-asyncstorage**, która pozwala na bezpieczne przechowywanie danych w zaszyfrowanej formie.
- Mechanizm szyfrowania zaimplementowano na poziomie aplikacji, dzięki czemu przechowywane notatki są chronione nawet w przypadku uzyskania dostępu do urządzenia przez osoby trzecie.

### **3. Bezpieczna komunikacja z serwerem**

Komunikacja między aplikacją a serwerem została zabezpieczona za pomocą:

- **Protokołu HTTPS**, zapewniającego szyfrowanie przesyłanych danych.
- Na potrzeby projektu korzystaliśmy z tunelowania za pomocą aplikacji ngrok

### **4. Zarządzanie uprawnieniami i bezpieczeństwo aplikacji**

Aplikacja została zaprojektowana z uwzględnieniem minimalizmu w zakresie żądanych uprawnień.

- Użytkownik jest proszony wyłącznie o niezbędne uprawnienia, takie jak dostęp do pamięci urządzenia w przypadku tworzenia kopii zapasowych.
- Dodano mechanizm **automatycznego blokowania aplikacji** po określonym czasie bezczynności, co zwiększa ochronę wrażliwych danych.

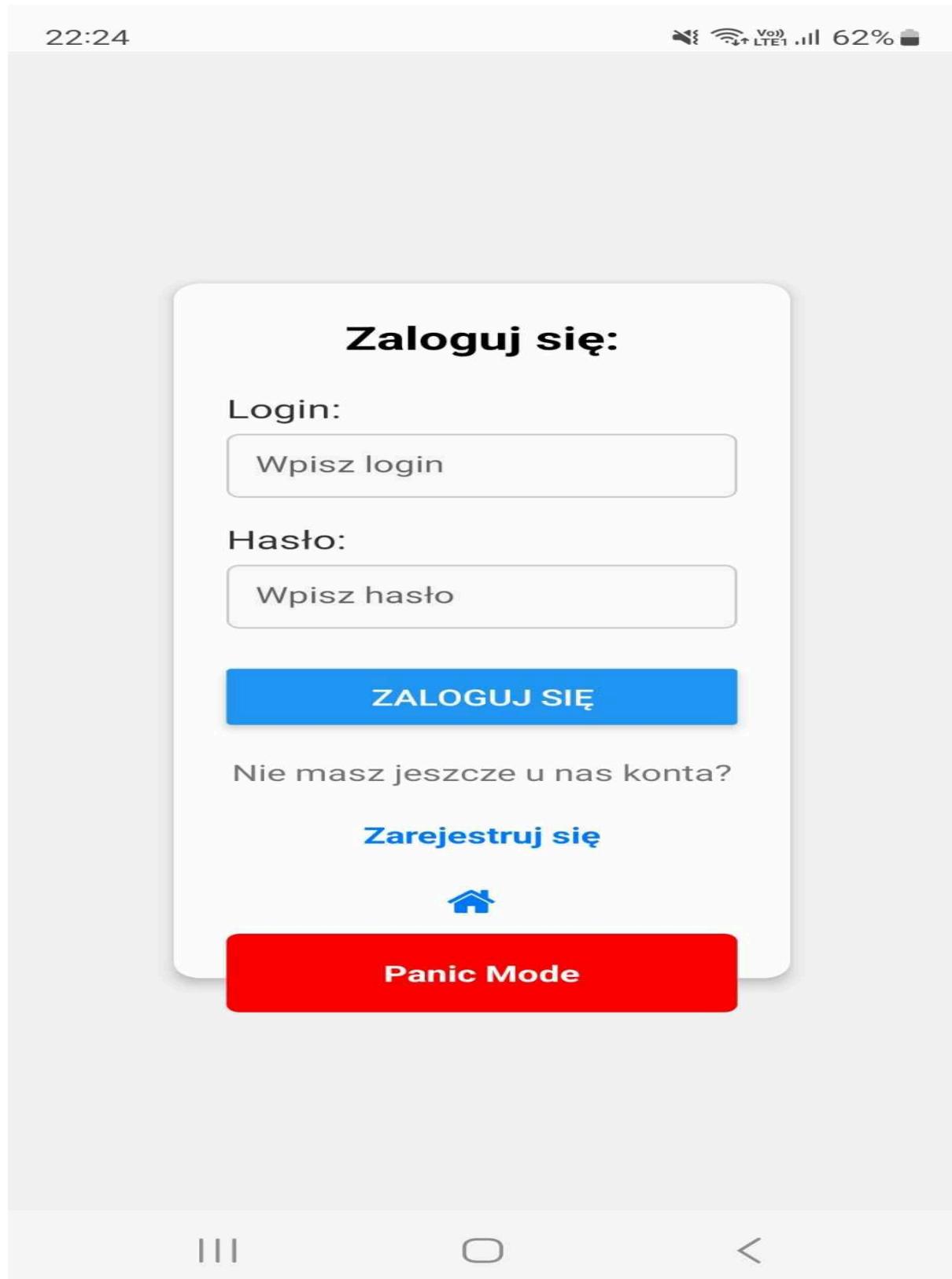
### **5. Mechanizmy zdalnego zarządzania danymi**

W przypadku zmiany urządzenia użytkownik ma możliwość:

- Tworzenia i przywracania kopii zapasowych w formie zaszyfrowanej, co pozwala na zachowanie integralności danych.

## 6. Testy manualne aplikacji

Po uruchomieniu aplikacji, uzyskujemy następujący widok:



Po naciśnięciu przycisku “Zarejestruj się”, uzyskujemy następujący widok:

The screenshot shows a mobile application interface for user registration. At the top, there is a header bar with the time "22:24", signal strength, battery level at "62%", and a camera icon. Below the header, the title "Zarejestruj się:" is displayed in a large, bold, black font. The form consists of five input fields, each with a label and a text input box. The first field is labeled "Podaj email:" and contains the value "rdawid238@gmail.com". The second field is labeled "Podaj imię:" and contains the value "Dawid". The third field is labeled "Podaj nazwisko:" and contains the value "Bodzon". The fourth field is labeled "Podaj hasło:" and contains the value ".....". The fifth field is labeled "Powtórz hasło:" and contains the value ".....". A large blue button at the bottom center contains the white text "Zarejestruj się". Below the button is a small blue house icon. At the very bottom of the screen, there are three grey navigation icons: three horizontal bars, a square, and a left arrow.

22:24

Zarejestruj się:

Podaj email:

rdawid238@gmail.com

Podaj imię:

Dawid

Podaj nazwisko:

Bodzon

Podaj hasło:

.....

Powtórz hasło:

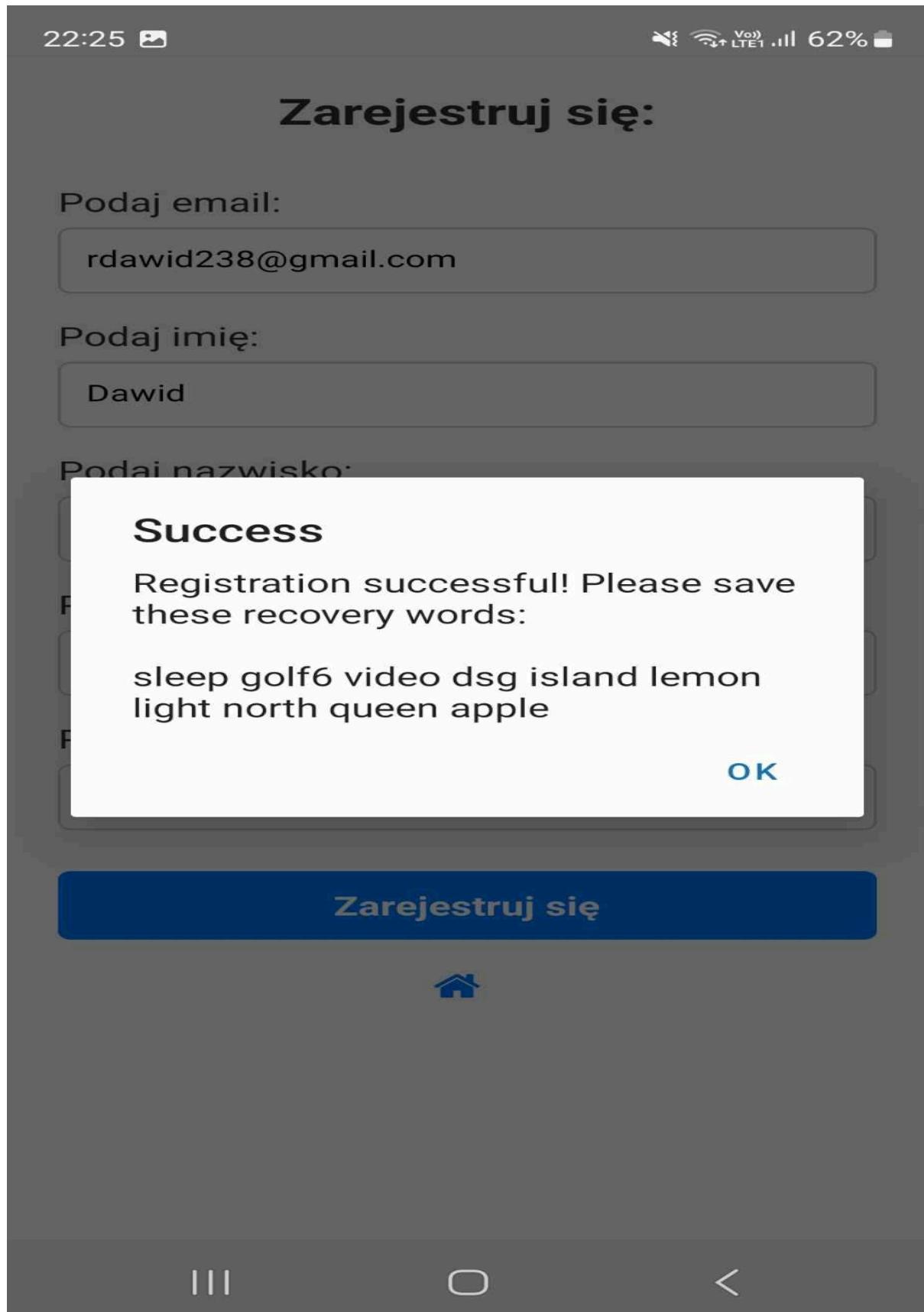
.....

Zarejestruj się

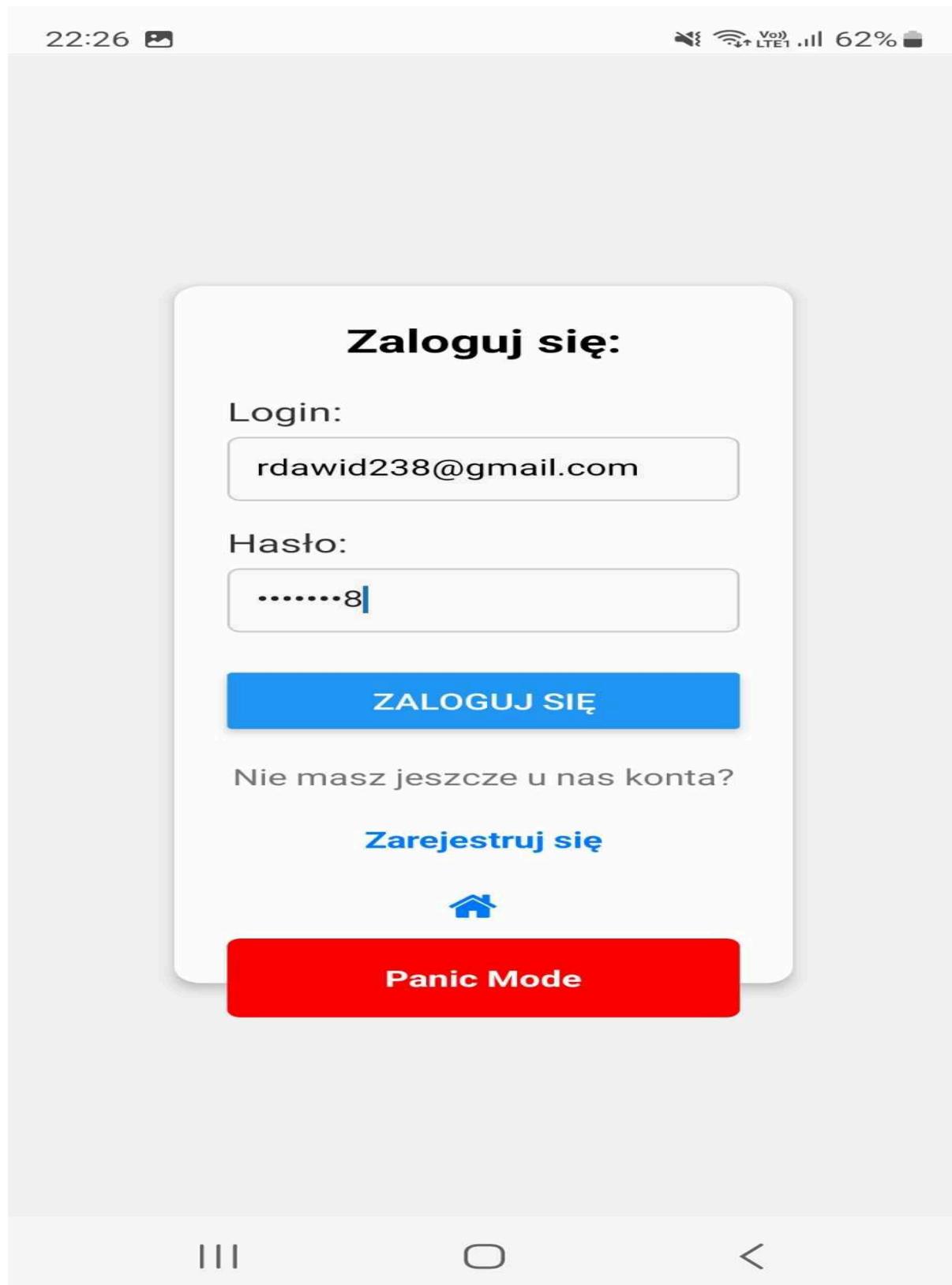
Home icon

Navigation icons: three horizontal bars, square, left arrow

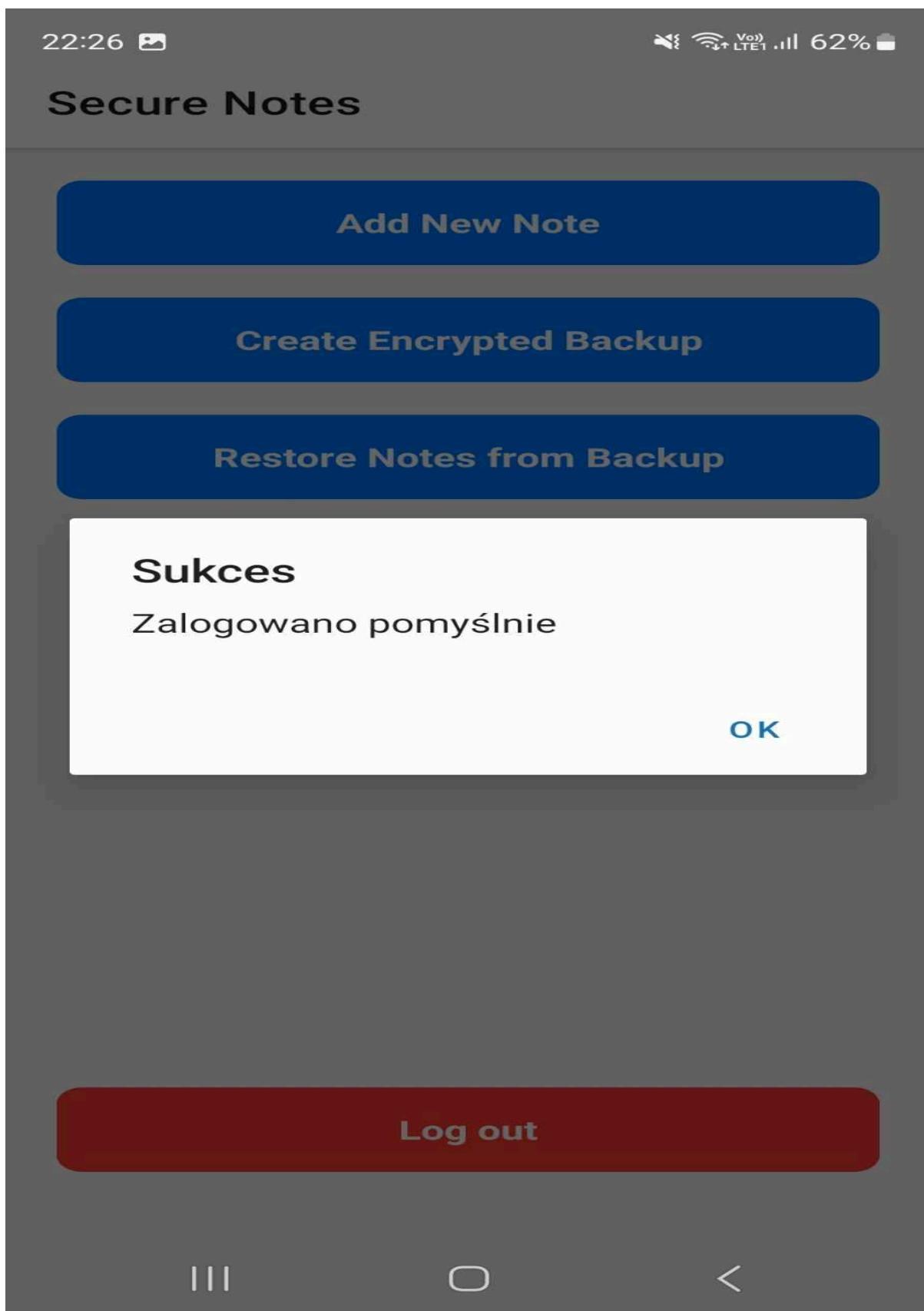
Po odpowiednim wypełnieniu pól i kliknięciu przycisku, dostajemy komunikat z informacją o rejestracji. Dodatkowo uzyskujemy słowa kluczowe, których ciąg będzie nam potrzebny w razie chęci zablokowania konta.



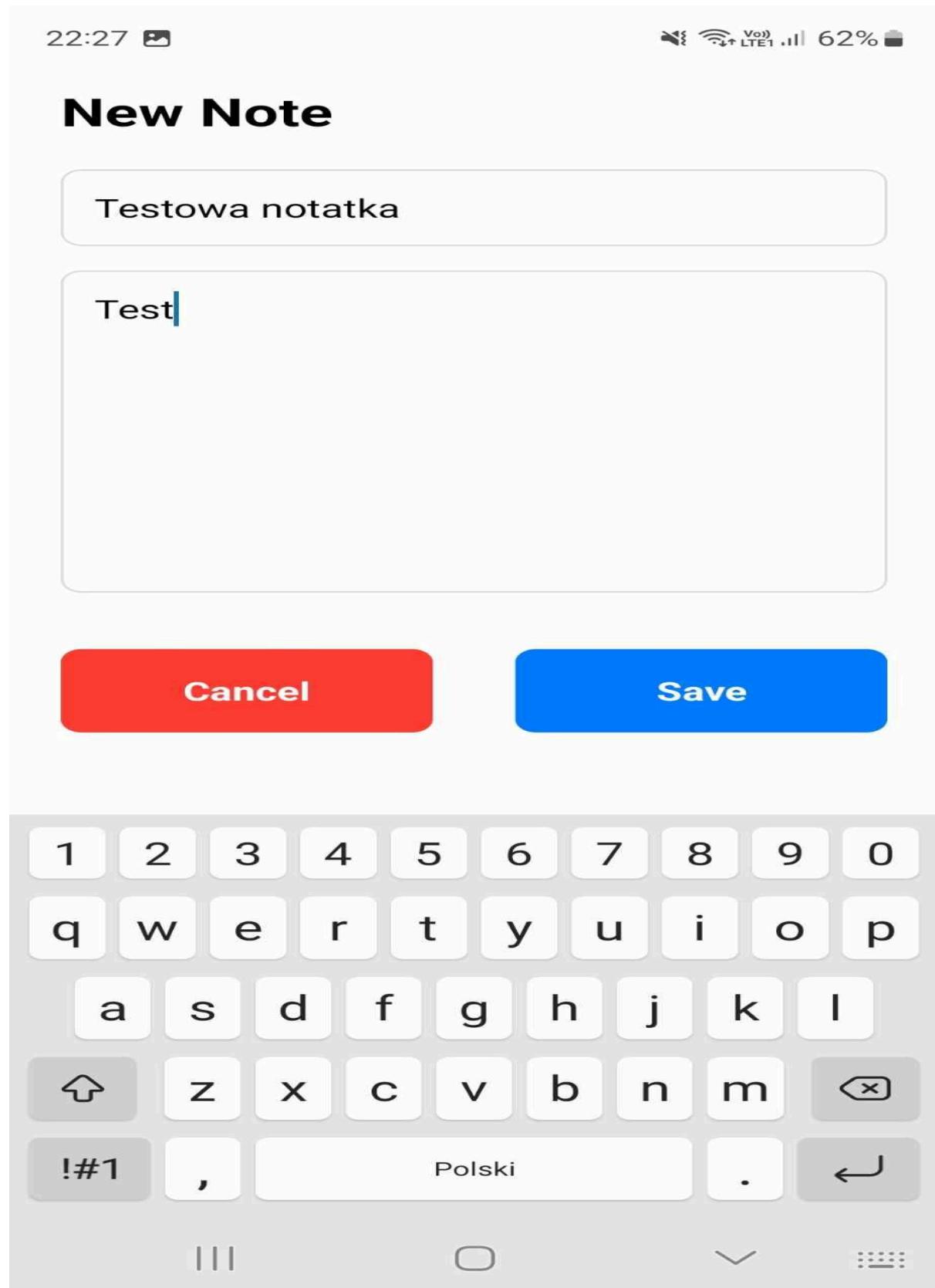
Po przejściu do zakładki odpowiedzialnej za logowanie po naciśnięciu przycisku "Zaloguj się", uzyskujemy taki widok.



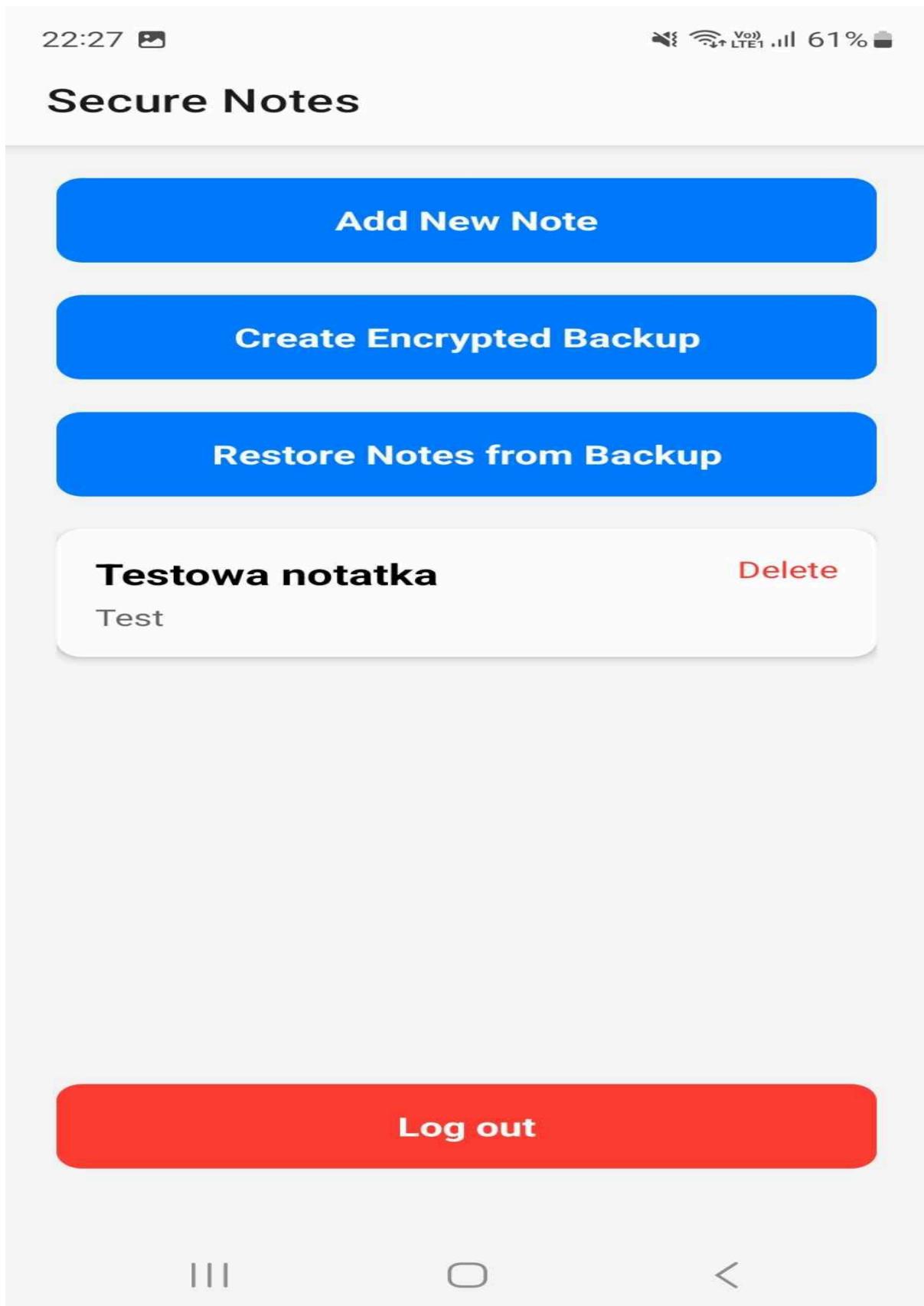
W przypadku poprawnego zalogowania, zostajemy przeniesieni do ekranu z notatkami.



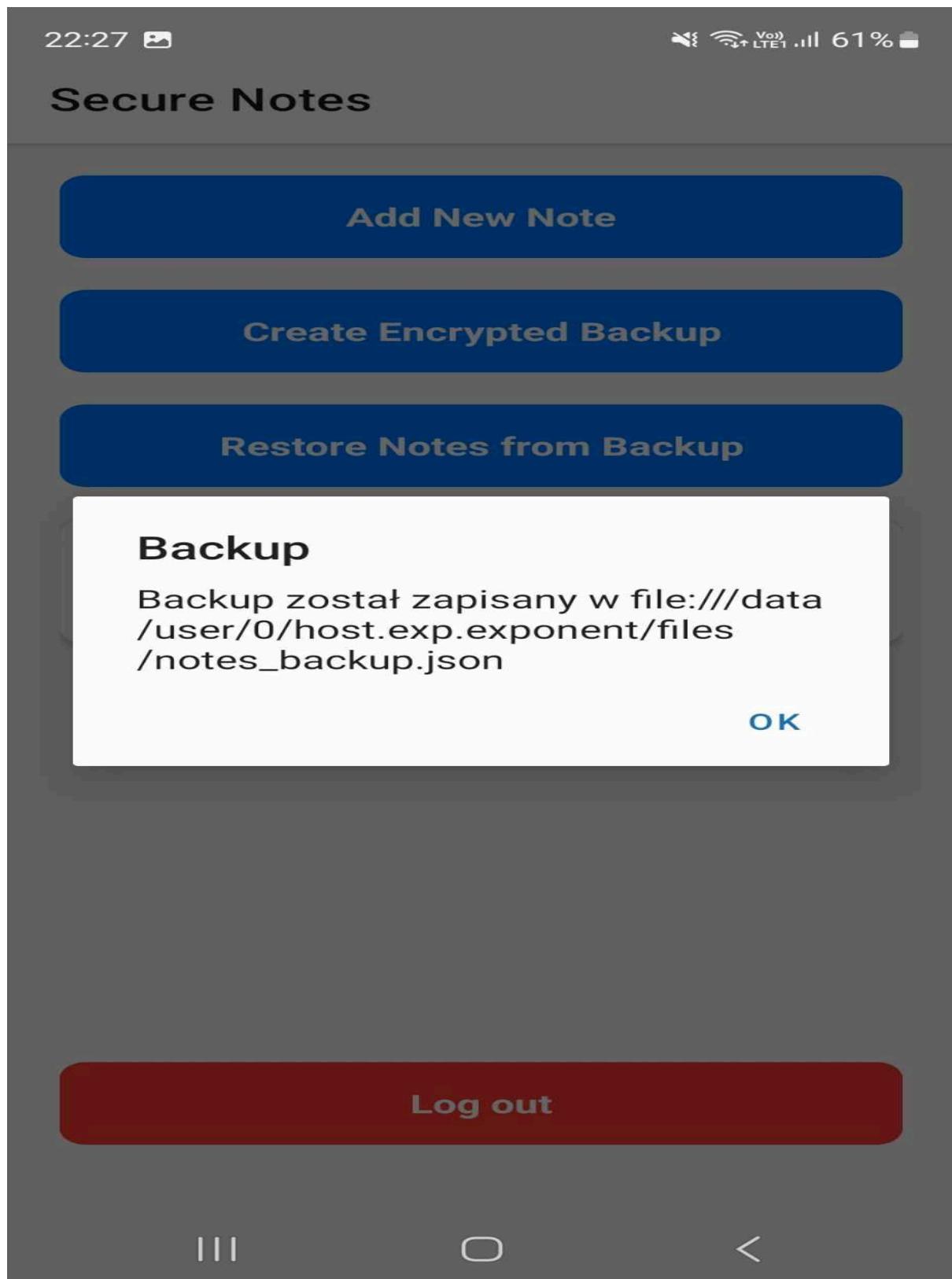
Po kliknięciu przycisku “Add New Note”, zostajemy przeniesieni do ekranu odpowiedzialnego za przygotowanie nowej notatki.



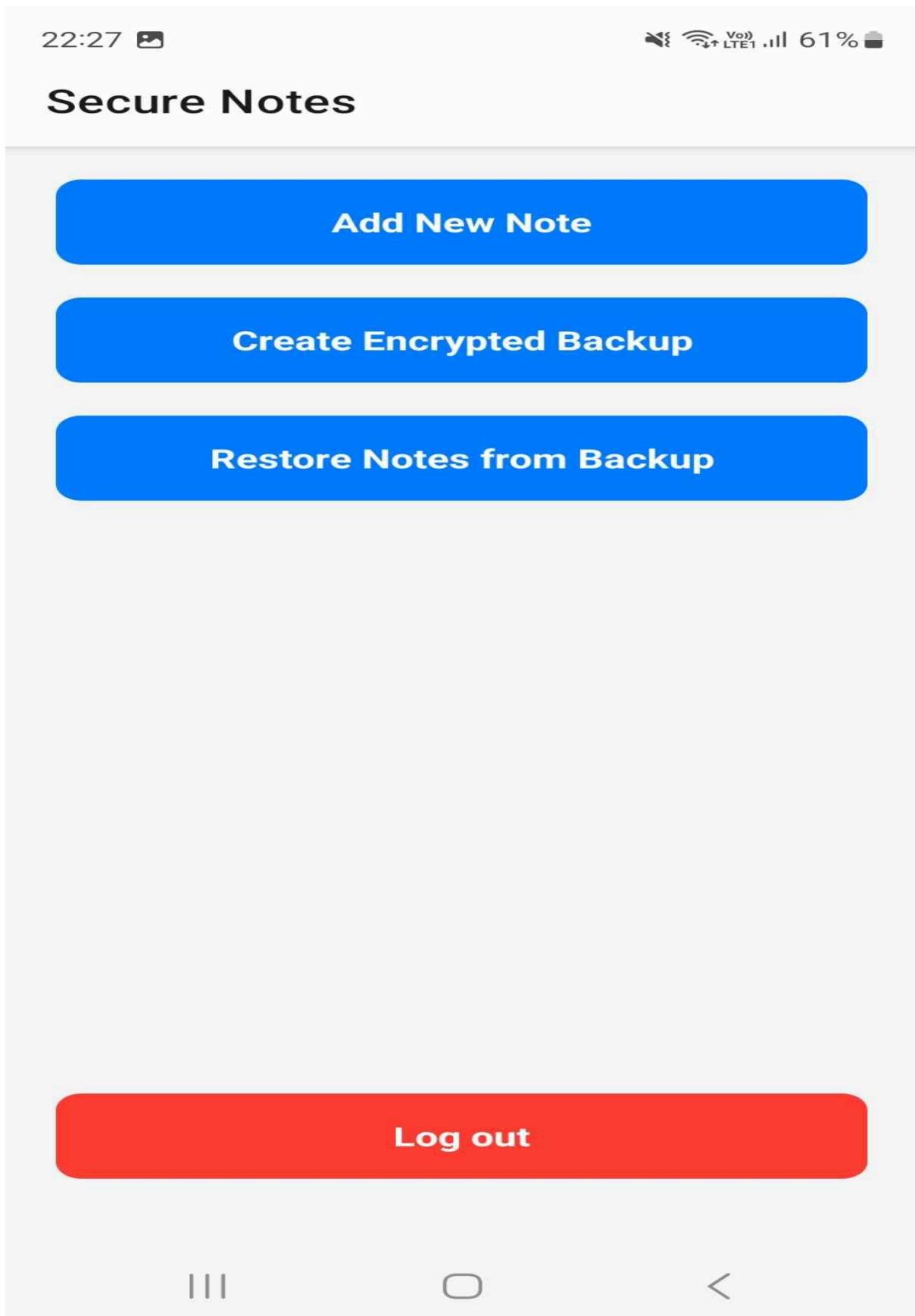
Jak widać, notatka została poprawnie dodana do naszej aplikacji.



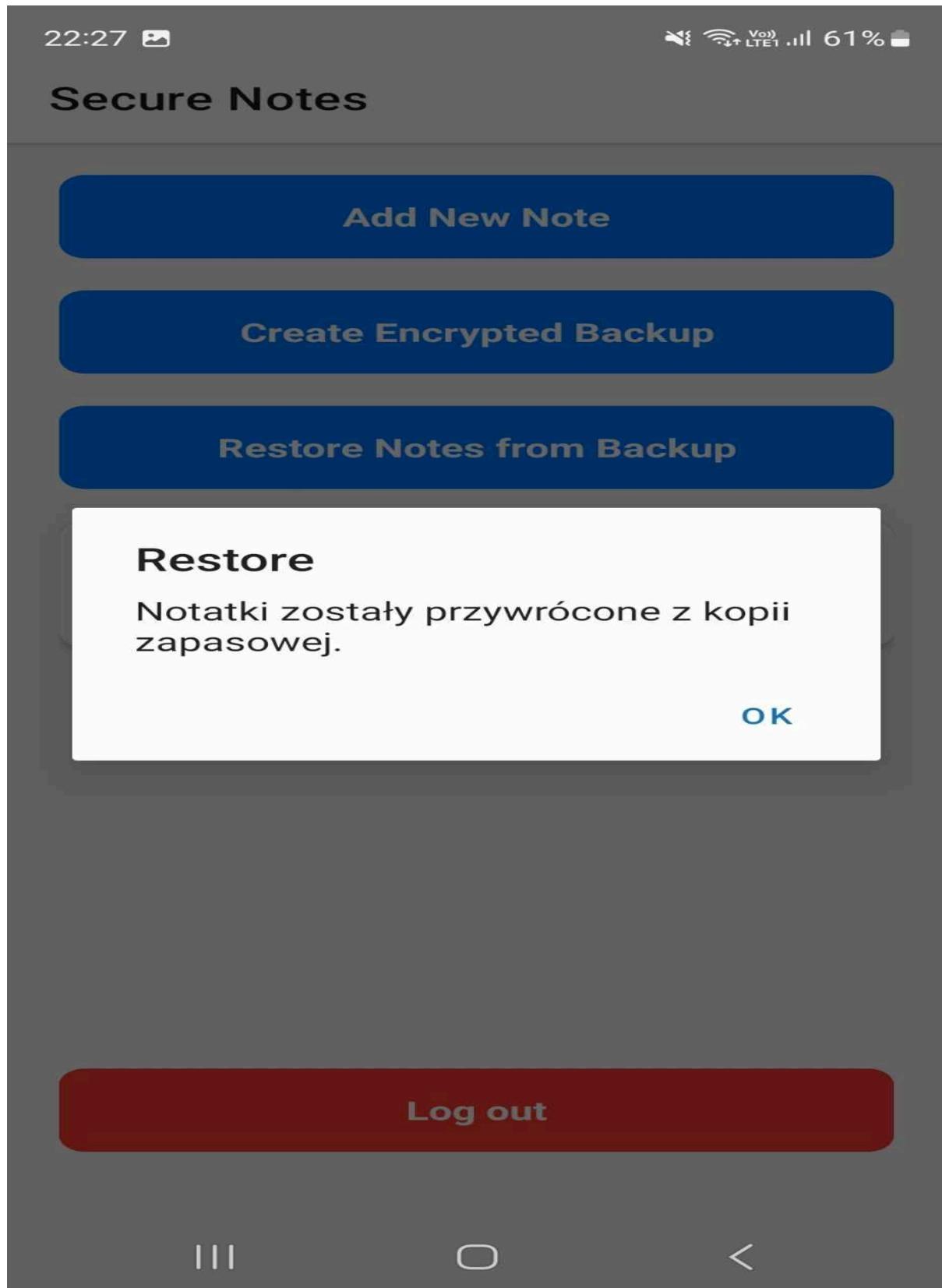
Po kliknięciu przycisku “Create Encrypted Backup”, zostaje utworzony backup naszych notatek.



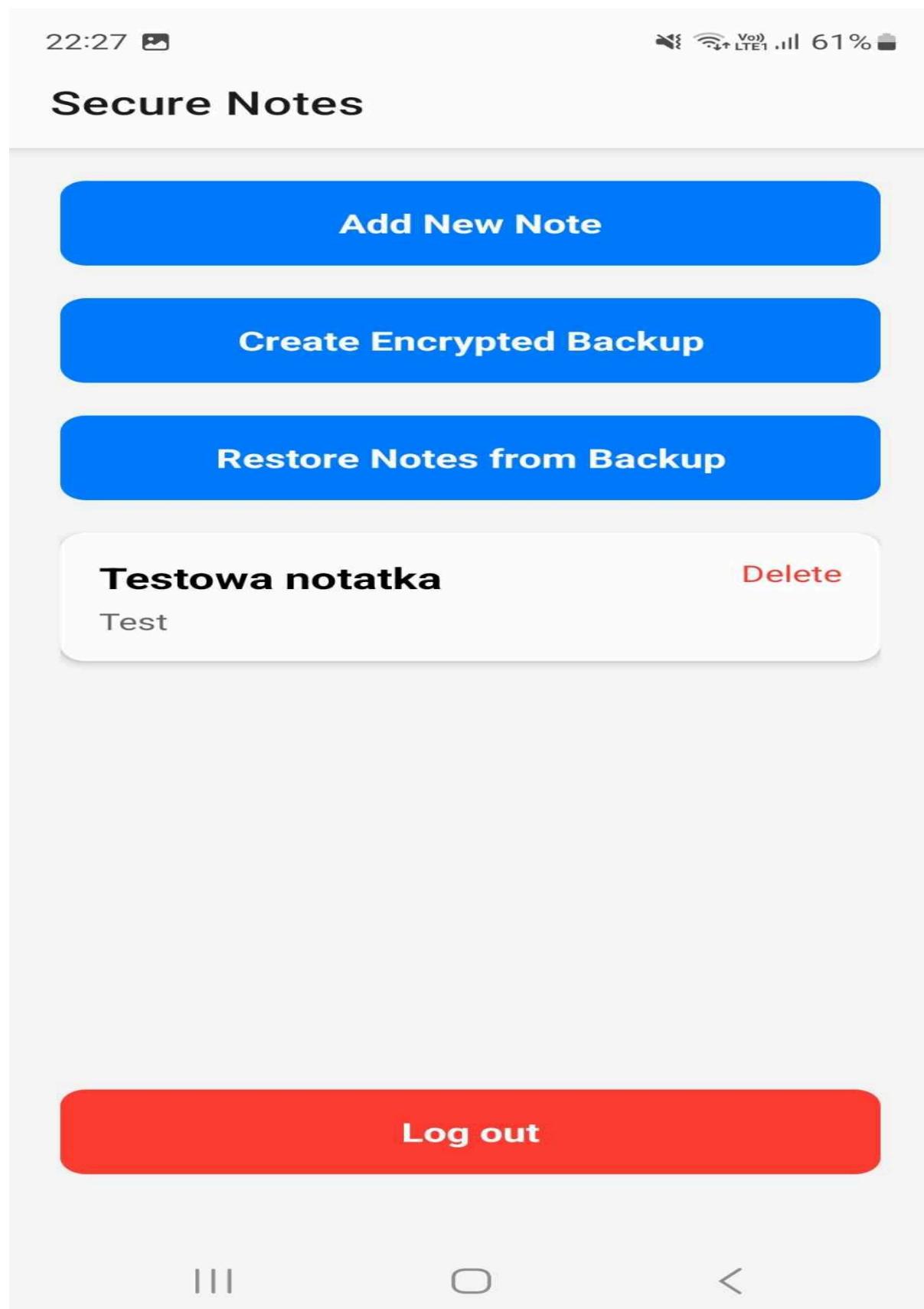
Usuwamy notatkę poprzez kliknięcie odpowiedniego przycisku.



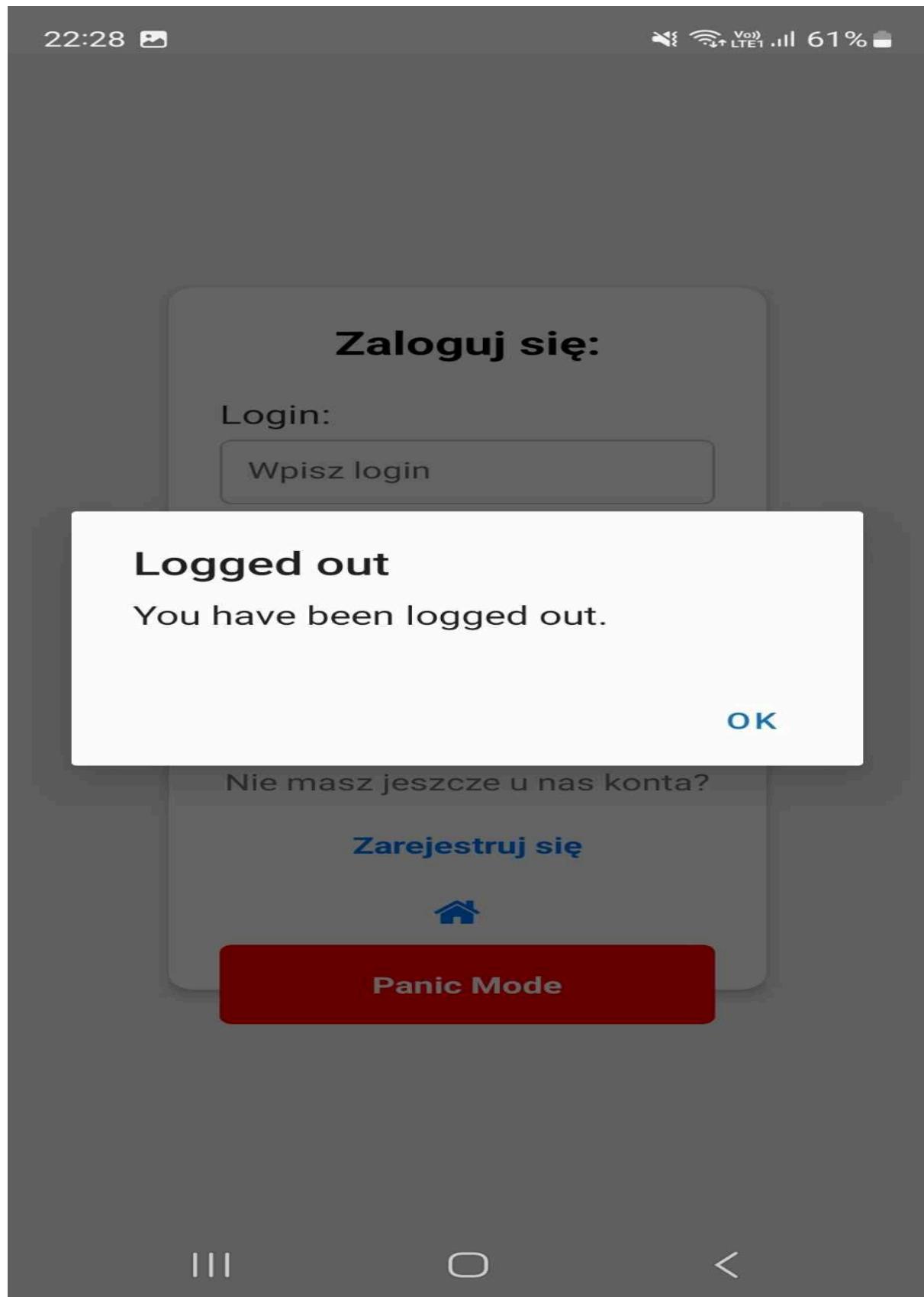
Po kliknięciu przycisku “Restore Notes from Backup” nasze notatki zostają przywrócone z powrotem do stanu, który został zapisany w backupie.



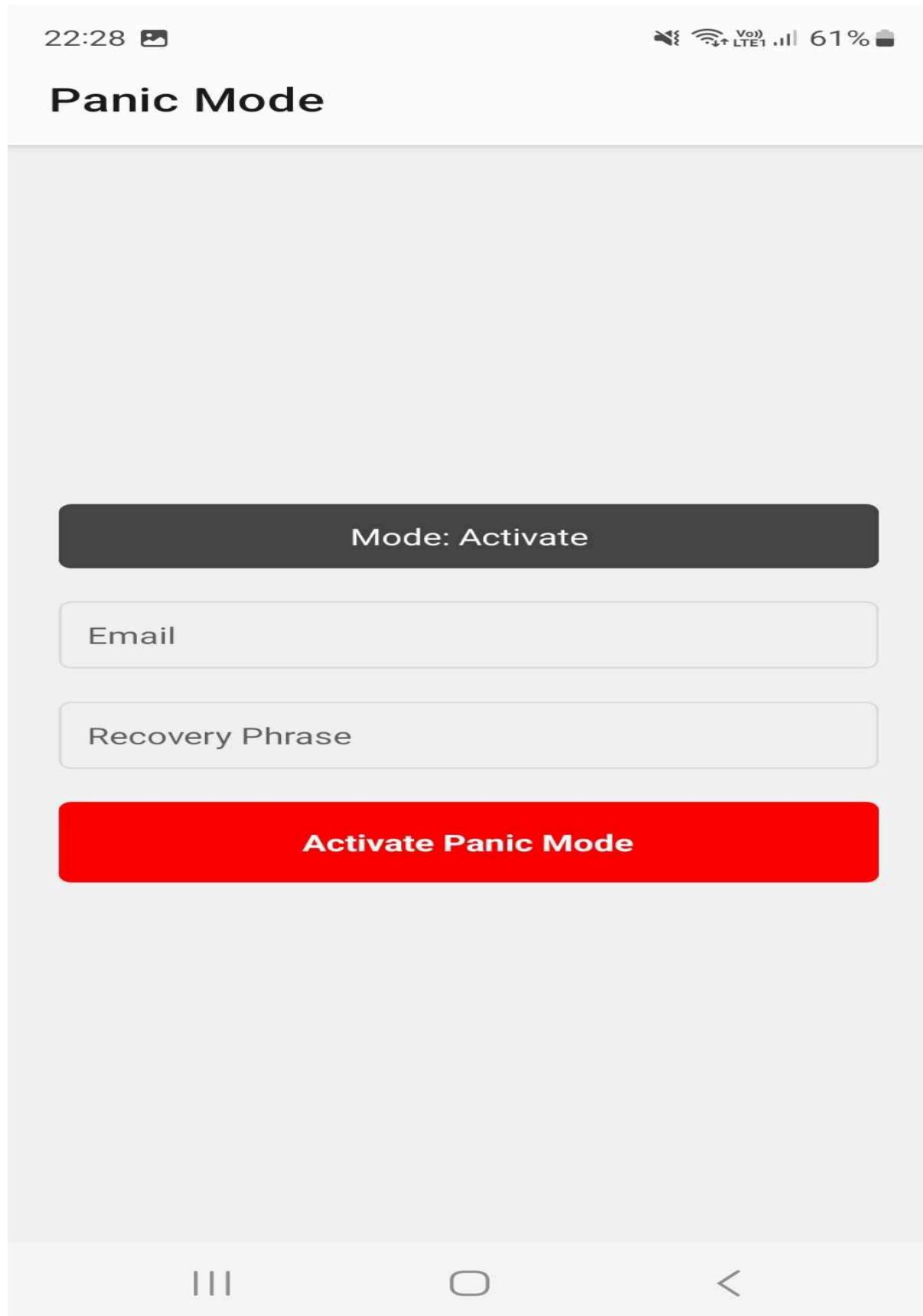
Jak widać, notatka wróciła z powrotem na swoje miejsce po załadowaniu backupu.



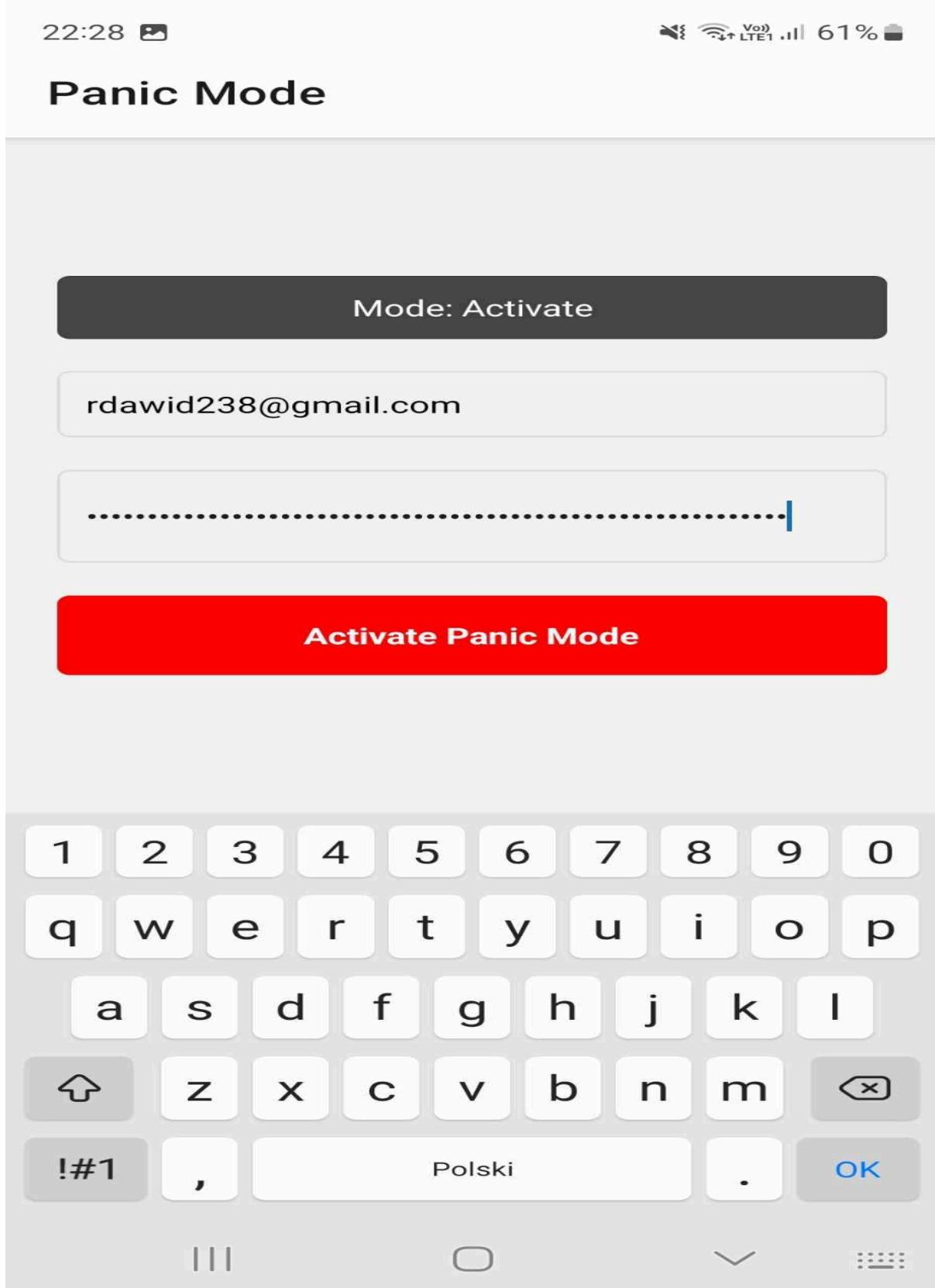
Po kliknięciu przycisku “Log out”, zostajemy poprawnie wylogowani o czym informuje nas komunikat.



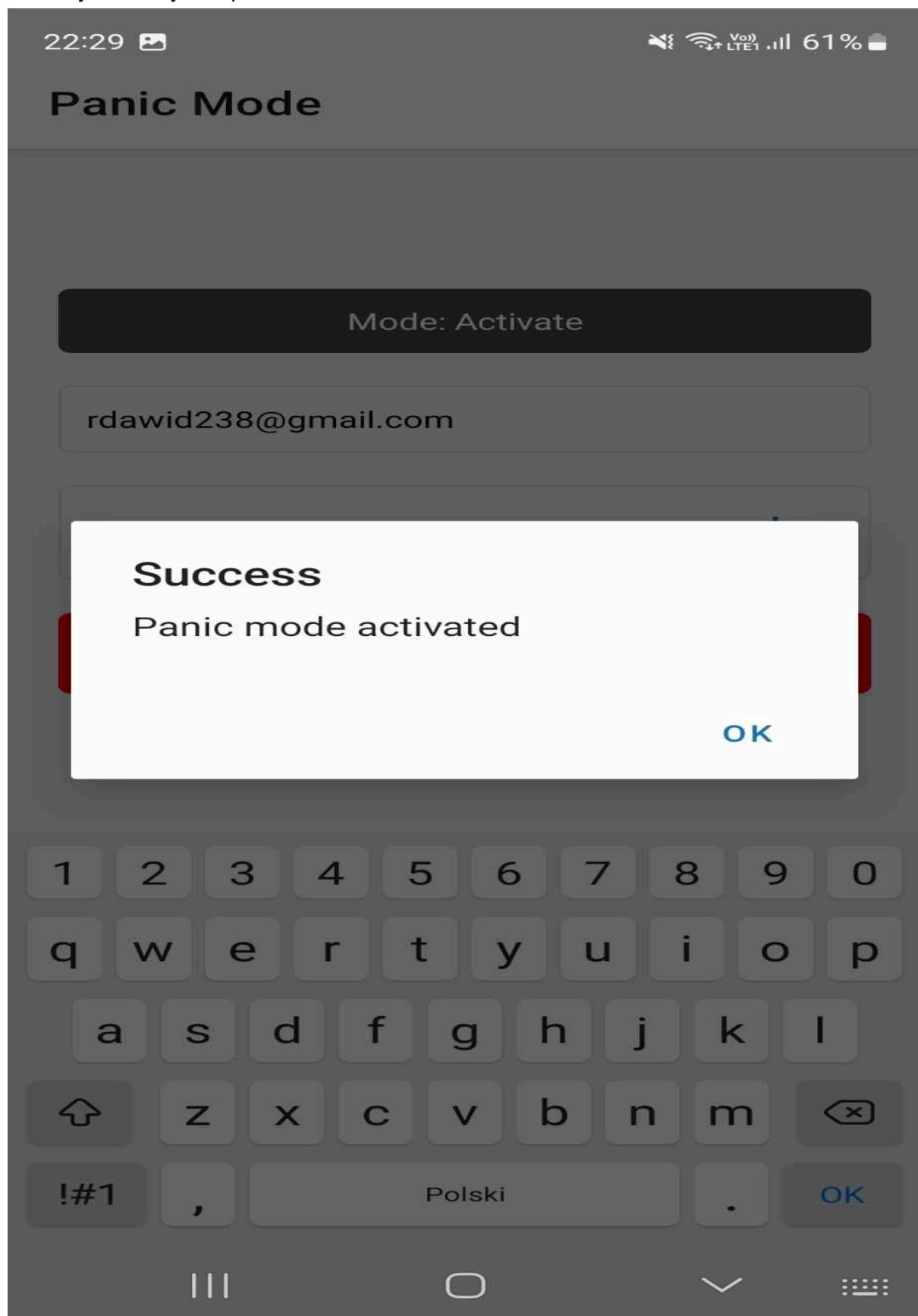
Po kliknięciu przycisku “Panic Mode”, zostajemy przeniesieni do następującego widoku.



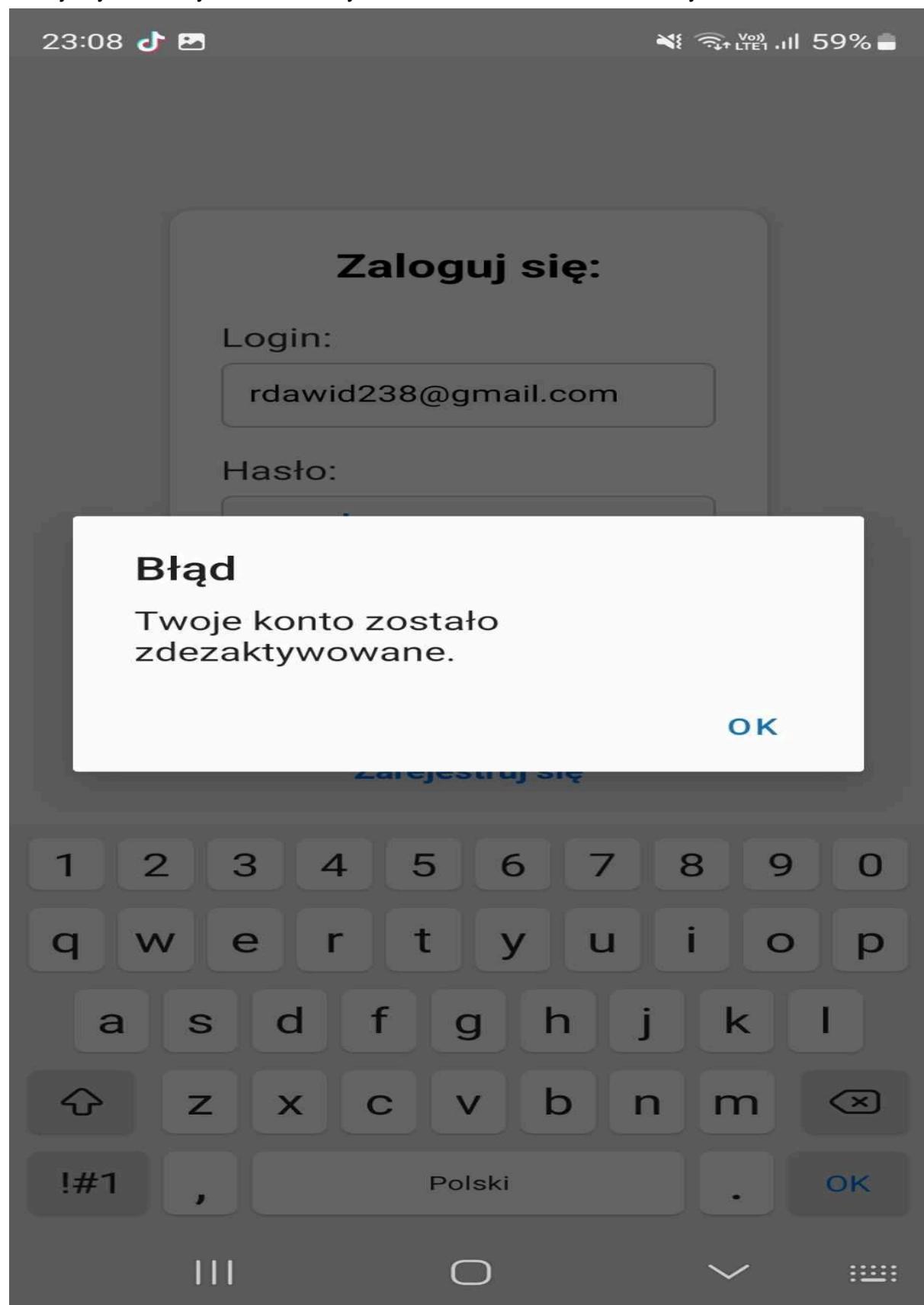
Możemy tutaj modyfikować status naszego konta. Jeśli Panic Mode jest włączony, dostęp do konta zostaje wyłączony. Aby go wyłączyć, musimy podać wygenerowany przy rejestracji ciąg wyrazów.



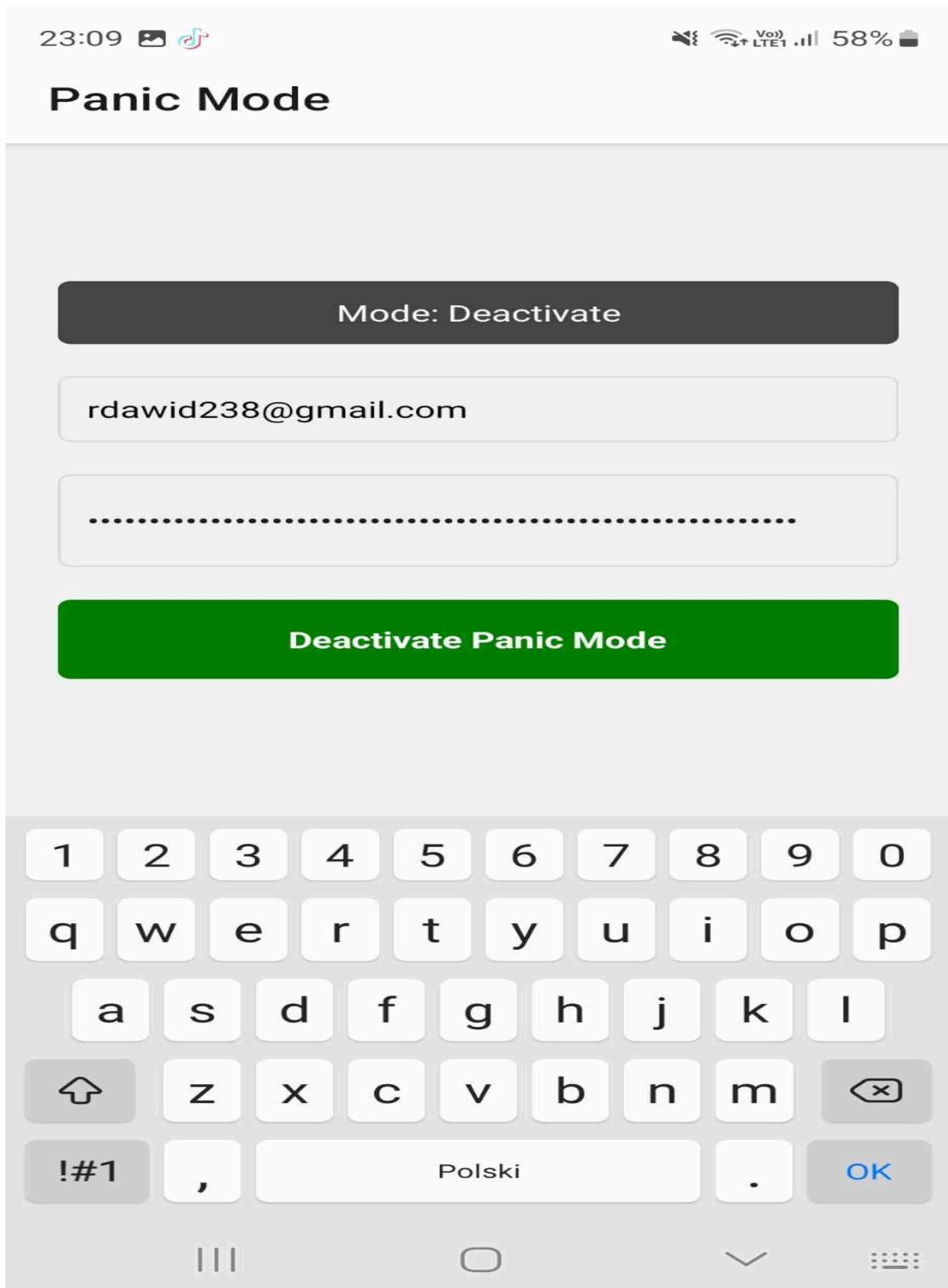
Po uzupełnieniu, "Panic mode" zostaje aktywowany, co wyłącza dostęp do naszego konta. Informuje nas o tym odpowiedni komunikat.



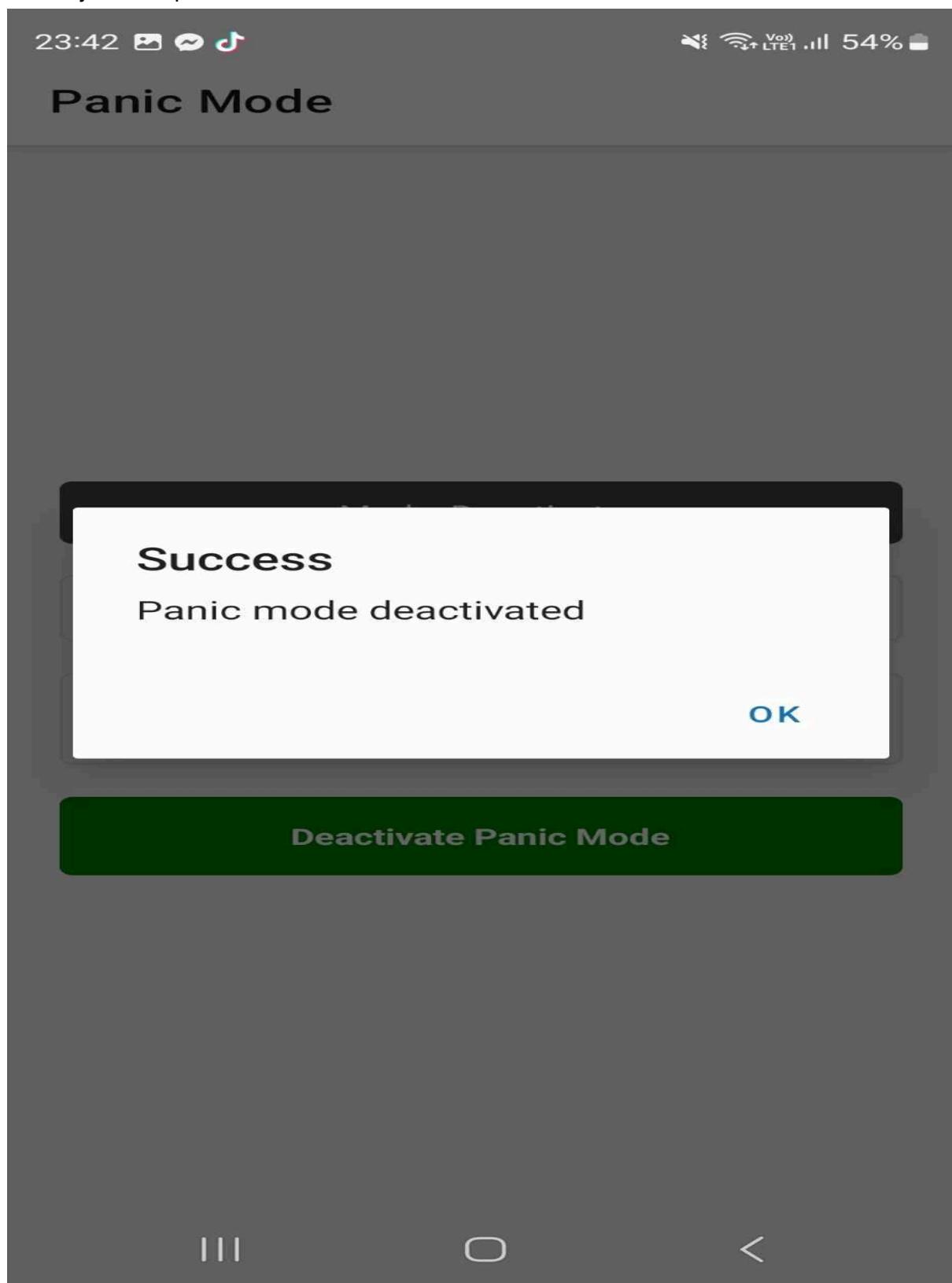
Teraz przy próbie zalogowania, nie zostajemy poprawnie zalogowani, a zamiast tego dostajemy stosowny komunikat o tym, że nasze konto zostało dezaktywowane.



Możemy oczywiście dezaktywować "Panic Mode" i tym samym przywrócić dostęp do naszego konta. W tym celu musimy zmienić zawartość Checkboxa poprzez naciśnięcie go i podanie odpowiednich danych.



Jak widać, aplikacja poprawnie zareagowała i konto z powrotem stało się aktywne, o czym informuje nas odpowiedni komunikat.



Teraz po poprawnym zalogowaniu, z powrotem mamy dostęp do aplikacji.

23:43 📲 💬 🎵

🔈 WiFi VoIP LTE 54% 📺

## Secure Notes

Add New Note

Create Encrypted Backup

Restore Notes from Backup

Log out



## 7. Omówienie najważniejszych sekcji kodu

### Frontend:

#### Funkcja handleLogin:

```
const handleLogin = async () => {
  if (!login || !password) {
    Alert.alert("Błąd", "Proszę wypełnić wszystkie pola");
    return;
  }

  setIsLoading(true);

  try {
    const response = await fetch(API_URL, {
      method: "POST",
      headers: {
        "Content-Type": "application/json",
      },
      body: JSON.stringify({
        email: login,
        password,
      }),
    });

    const data = await response.json();

    if (response.ok) {
      try {
        await storeAuthData(data.token);
        const userId = await SecureStore.getItemAsync("userId");
        const isPanicActive = await getPanicStatus(userId!);
        if (isPanicActive) {
          Alert.alert("Błąd", data.message || "Twoje konto zostało zdezaktywowane.");
        } else {
          Alert.alert("Sukces", "Zalogowano pomyślnie");
          router.push("/notes");
        }
      } catch {
        Alert.alert("Błąd", data.message || "Twoje konto zostało zdezaktywowane.");
      }
    } else {
      Alert.alert("Błąd", data.message || "Wystąpił błąd podczas logowania");
    }
  } catch (error) {
    Alert.alert(
      "Błąd",
      "Nie można połączyć się z serwerem. Sprawdź połączenie internetowe."
    );
  } finally {
    setLoading(false);
  }
};
```

Funkcja **handleLogin** najpierw sprawdza, czy użytkownik wypełnił pola login i hasło, a jeśli nie, wyświetla komunikat o błędzie. Następnie wysyła dane logowania do backendu metodą POST i przetwarza odpowiedź. W przypadku sukcesu zapisuje token w SecureStore, sprawdza status konta w trybie Panic Mode i, jeśli konto jest aktywne, przekierowuje użytkownika na ekran notatek. W razie błędów (np. dezaktywowane konto lub problemy z połączeniem) wyświetla odpowiednie alerty. Cały proces obsługuje stan ładowania za pomocą flagi isLoading.

### Funkcja handleRegister:

```
const handleRegister = async () => {
  if (!validateForm()) return;

  setIsLoading(true);
  try {
    const response = await fetch(API_URL, {
      method: "POST",
      headers: {
        "Content-Type": "application/json",
      },
      body: JSON.stringify(formData),
    });

    const data = await response.json();

    if (!response.ok) {
      throw new Error("Registration failed");
    }

    Alert.alert(
      "Success",
      `Registration successful! Please save these recovery words:\n\n${data.recoveryPhrase}`,
      [
        {
          text: "OK",
          onPress: () => router.replace("/login"),
        },
      ],
    );
  } catch (error) {
    Alert.alert(
      "Error",
      error instanceof Error ? error.message : "Registration failed"
    );
  } finally {
    setIsLoading(false);
  }
};
```

Funkcja **handleRegister** weryfikuje poprawność formularza za pomocą validateForm, a następnie wysyła dane rejestracyjne do backendu metodą POST. Jeśli rejestracja zakończy się sukcesem, wyświetla alert z frazą odzyskiwania danych, prosząc użytkownika o jej zapisanie, po czym przekierowuje na ekran logowania. W przypadku błędu wyświetla komunikat o problemie z rejestracją. Proces jest obsługiwany z użyciem wskaźnika ładowania isLoading, który zapobiega wielokrotnemu wysyłaniu żądania w trakcie przetwarzania.

### Funkcja handleCreateBackup:

```
const handleCreateBackup = async () => {
  try {
    setIsLoading(true);
    const backupFileName = "notes_backup.json";
    const backupFilePath = `${FileSystem.documentDirectory}${backupFileName}`;

    const encryptedNotes = notes.map((note) => ({
      ...note,
      content: CryptoJS.AES.encrypt(note.content, ENCRYPTION_KEY).toString(),
    }));
    const backupContent = JSON.stringify(encryptedNotes);

    await FileSystem.writeStringAsync(backupFilePath, backupContent, {
      encoding: FileSystem.EncodingType.UTF8,
    });

    Alert.alert("Backup", `Backup został zapisany w ${backupFilePath}`);
  } catch (error) {
    Alert.alert("Error", "Nie udało się utworzyć kopii zapasowej.");
  } finally {
    setIsLoading(false);
  }
};
```

Funkcja **handleCreateBackup** tworzy kopię zapasową notatek, szyfrując ich treść za pomocą CryptoJS.AES i zapisując zaszyfrowane dane w pliku notes\_backup.json w katalogu dokumentów aplikacji. Jeśli operacja zakończy się sukcesem, wyświetla komunikat z lokalizacją pliku kopii. W przypadku błędu wyświetla alert z informacją o niepowodzeniu.

### Funkcja handleRestoreBackup:

```
const handleRestoreBackup = async () => {
  try {
    setIsLoading(true);
    const backupFileName = "notes_backup.json";
    const backupFilePath = `${FileSystem.documentDirectory}${backupFileName}`;

    const fileInfo = await FileSystem.getInfoAsync(backupFilePath);
    if (!fileInfo.exists) {
      Alert.alert("Restore", "Nie znaleziono kopii zapasowej.");
      return;
    }

    const backupContent = await FileSystem.readAsStringAsync(backupFilePath, {
      encoding: FileSystem.EncodingType.UTF8,
    });

    const decryptedNotes = JSON.parse(backupContent).map((note: any) => ({
      ...note,
      content: CryptoJS.AES.decrypt(note.content, ENCRYPTION_KEY).toString(
        CryptoJS.enc.Utf8
      ),
    }));
    setNotes(decryptedNotes);
    Alert.alert("Restore", "Notatki zostały przywrócone z kopii zapasowej.");
  } catch (error) {
    Alert.alert("Error", "Nie udało się przywrócić notatek.");
  } finally {
    setIsLoading(false);
  }
};
```

Funkcja **handleRestoreBackup** przywraca notatki z kopii zapasowej zapisanej w notes\_backup.json. Sprawdza istnienie pliku, odczytuje jego zawartość i deszyfruje dane, przywracając je do stanu czytelnego. Jeśli kopia nie istnieje lub operacja się nie powiedzie, wyświetla odpowiedni komunikat. Obie funkcje zarządzają stanem ładowania za pomocą setLoading.

### Funkcja activatePanicMode:

```
const activatePanicMode = async (payload: PanicModePayload) => {
  setIsLoading(true);
  setError(null);
  try {
    const response = await fetch(`#${API_URL}/panic-mode`, {
      method: "POST",
      headers: {
        "Content-Type": "application/json",
      },
      body: JSON.stringify(payload),
    });
    const data = await response.json();
    return data;
  } catch (err) {
    setError(err instanceof Error ? err.message : "An error occurred");
    throw err;
  } finally {
    setIsLoading(false);
  }
};
```

Funkcja **activatePanicMode** aktywuje tryb paniczny, wysyłając dane (payload) do serwera w celu uruchomienia tego trybu. Funkcja wykonuje zapytanie HTTP typu POST do endpointu /panic-mode, przesyłając dane w formacie JSON. Po otrzymaniu odpowiedzi z serwera, zwraca dane w formacie JSON. Jeśli wystąpi błąd, wyświetla komunikat o błędzie. Po zakończeniu operacji zmienia stan ładowania na false.

### Funkcja deactivatePanicMode:

```
const deactivatePanicMode = async (payload: PanicModePayload) => {
  setIsLoading(true);
  setError(null);
  try {
    const response = await fetch(`#${API_URL}/panic-mode/reverse`, {
      method: "POST",
      headers: {
        "Content-Type": "application/json",
      },
      body: JSON.stringify(payload),
    });
    const data = await response.json();
    return data;
  } catch (err) {
    setError(err instanceof Error ? err.message : "An error occurred");
    throw err;
  } finally {
    setIsLoading(false);
  }
};
```

Funkcja **deactivatePanicMode** dezaktywuje tryb paniczny, wysyłając dane (payload) do serwera w celu jego wyłączenia. Funkcja wykonuje zapytanie HTTP typu POST do endpointu /panic-mode/reverse, przesyłając dane w formacie JSON. Po otrzymaniu odpowiedzi z serwera, zwraca dane w formacie JSON. Jeśli wystąpi błąd, wyświetla komunikat o błędzie. Po zakończeniu operacji zmienia stan ładowania na false.

### Funkcja `createNote`:

```
const createNote = async (note: Note, encryptionKey: string) => {
  try {
    const encryptedContent = encrypt(note.content, encryptionKey);
    const response = await fetch(API_URL, {
      method: "POST",
      headers: {
        "Content-Type": "application/json",
      },
      body: JSON.stringify([
        ...note,
        content: encryptedContent,
      ]),
    });
    return await response.json();
  } catch (error) {
    console.error("Error creating note:", error);
    throw error;
  }
};
```

Funkcja `createNote` tworzy nową notatkę, szyfrując jej zawartość przed wysłaniem na serwer. Przyjmuje dwa argumenty: obiekt `note` zawierający dane notatki oraz `encryptionKey` do szyfrowania treści. Najpierw funkcja szyfruje zawartość notatki za pomocą funkcji `encrypt`, a następnie wysyła zapytanie HTTP typu POST do serwera, przekazując zaszyfrowaną treść oraz inne dane notatki w formacie JSON. Po otrzymaniu odpowiedzi z serwera, zwraca wynik w formacie JSON. W przypadku błędu, np. problemów z połączeniem lub odpowiedzią serwera, funkcja łąpie wyjątek, loguje go i przekazuje dalej.

## Backend:

### Endpoint login:

```
@Post(EndPoints.login)
@ApiOperation({ summary: 'Login a user' })
@ApiBody({ type: LoginPayload })
@ApiResponse({
  status: 200,
  description: 'User logged in successfully',
  type: MessageResponse,
})
@ApiResponse({ status: 400, description: 'Invalid credentials' })
async login(@Body() payload: LoginPayload, @Res() res: Response) {
  const { token, user } = await this.authService.login(payload);
  res.cookie(Cookies.token, token, {
    httpOnly: true,
    maxAge: Number(process.env.EXPIRE_TIME),
  });
  return res.status(HttpStatus.OK).json({ user, token });
}
```

Endpoint **login** obsługuje logowanie użytkownika. Przyjmuje dane logowania w postaci obiektu LoginPayload w ciele zapytania i wykorzystuje serwis authService.login do przeprowadzenia logowania, zwracając token oraz dane użytkownika. Po pomyślnym logowaniu, token jest zapisywany w ciasteczku o nazwie Cookies.token, z opcją httpOnly oraz czasem wygasania pobranym z zmiennej środowiskowej EXPIRE\_TIME. Endpoint zwraca odpowiedź w formacie JSON z danymi użytkownika oraz tokenem, ustawiając status odpowiedzi na 200 OK. W przypadku nieprawidłowych danych logowania, zwróci odpowiedź z kodem 400 oraz odpowiednią wiadomością o błędzie.

### Endpoint register:

```
@Post(EndPoints.register)
@ApiOperation({ summary: 'Register a new user' })
@ApiBody({ type: RegisterPayload })
@ApiResponse({
  status: 201,
  description: 'User registered successfully',
  schema: {
    properties: {
      message: { type: 'string' },
      recoveryPhrase: { type: 'string' },
    },
  },
})
@ApiResponse({ status: 400, description: 'Invalid credentials' })
@ApiResponse({ status: 409, description: 'User already exists' })
async register(@Body() payload: RegisterPayload) {
  const user = await this.authService.register(payload);
  return {
    message: 'User registered successfully',
    recoveryPhrase: user.recoveryPhrase,
  };
}
```

Endpoint **register** obsługuje rejestrację nowego użytkownika. Przyjmuje dane rejestracyjne w postaci obiektu RegisterPayload w ciele zapytania i wykorzystuje serwis authService.register do zarejestrowania użytkownika. Po pomyślnym zarejestrowaniu, endpoint zwraca odpowiedź w formacie JSON z wiadomością "User registered successfully" oraz frazą odzyskiwania (recovery phrase) przypisaną do nowo utworzonego użytkownika. Endpoint ustawia status odpowiedzi na 201 Created. W przypadku błędnych danych rejestracyjnych, zwróci odpowiedź z kodem 400 oraz odpowiednią wiadomością o błędzie, a jeśli użytkownik o podanych danych już istnieje, zwróci odpowiedź z kodem 409 oraz informacją o konflikcie (użytkownik już istnieje).

### Endpoint backup:

```
@Post('backup')
@HttpCode(HttpStatus.CREATED)
@ApiOperation({ summary: 'Create an encrypted backup of all notes' })
@ApiResponse({
  status: 201,
  description: 'Backup has been successfully created.',
})
@ApiResponse({ status: 500, description: 'Failed to create backup.' })
async createBackup() {
  await this.notesService.encryptedBackup();
}
```

Endpoint **backup** służy do tworzenia zaszyfrowanej kopii zapasowej wszystkich notatek. Po wywołaniu endpointu, funkcja encryptedBackup w serwisie notesService generuje kopię zapasową notatek, szyfrując jej zawartość. Jeśli operacja zakończy się sukcesem, endpoint zwraca odpowiedź z kodem statusu 201 Created oraz komunikatem, że kopia została pomyślnie utworzona. W przypadku niepowodzenia, zwraca odpowiedź z kodem 500 i informacją o błędzie w trakcie tworzenia kopii zapasowej.

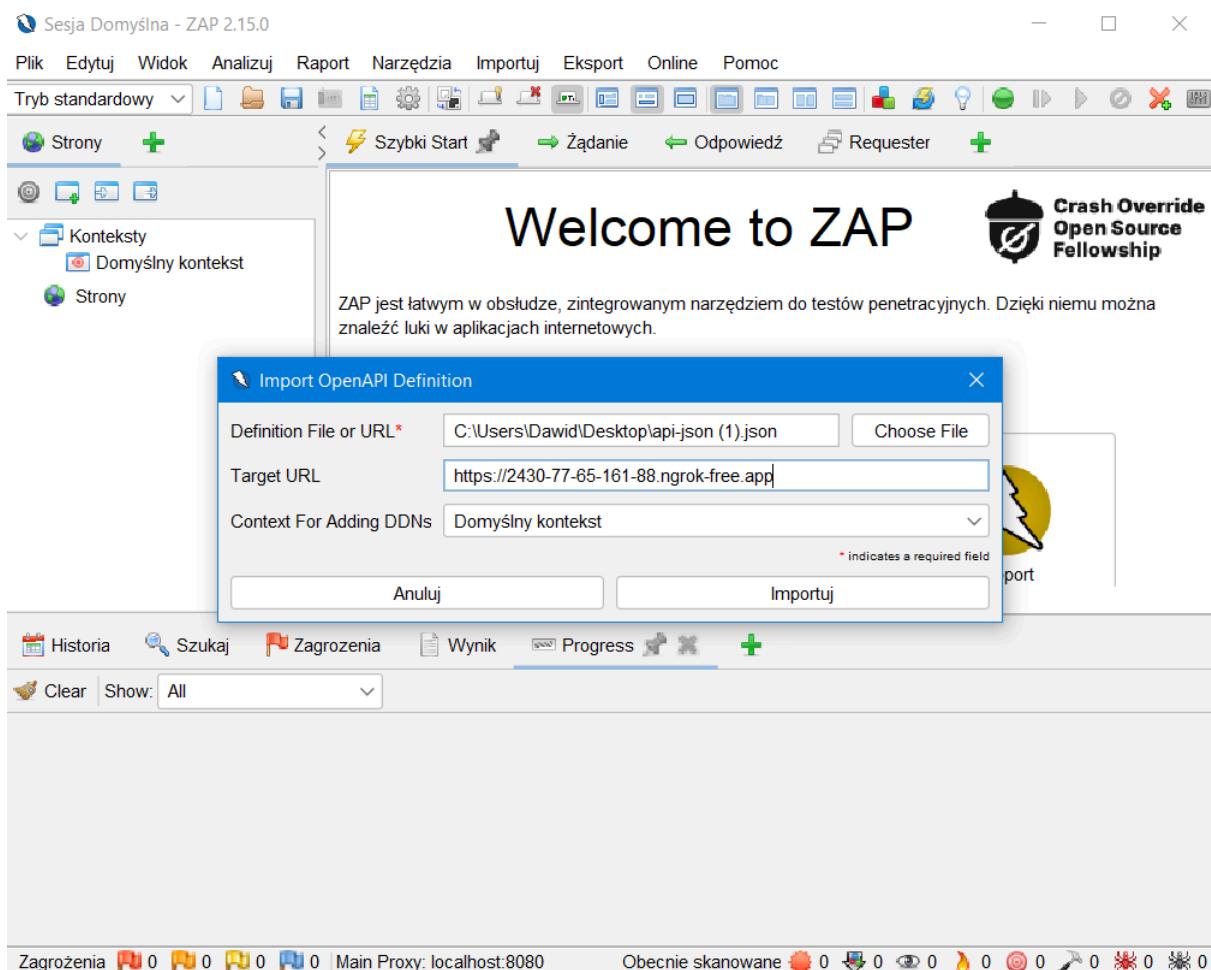
### Endpoint restore:

```
@Post('restore')
@HttpCode(HttpStatus.OK)
@ApiOperation({ summary: 'Restore notes from a backup file' })
async restoreBackup(@Body('filePath') filePath: string) {
  await this.notesService.restoreFromBackup(filePath);
}
```

Endpoint **restore** służy do przywracania notatek z pliku kopii zapasowej. Przyjmuje w ciele zapytania ścieżkę do pliku kopii zapasowej (filePath). Funkcja restoreFromBackup w serwisie notesService używa tej ścieżki do odczytania pliku i przywrócenia zapisanych w nim notatek. Po pomyślnym przywróceniu notatek, endpoint zwraca odpowiedź z kodem statusu 200 OK.

## **8. Testowanie bezpieczeństwa**

Testy aplikacji wykonane przy pomocy programu ZAP:



The screenshot shows the ZAP 2.15.0 interface. The top navigation bar includes 'Sesja Domyslna - ZAP 2.15.0', 'Plik', 'Edytuj', 'Widok', 'Analizuj', 'Raport', 'Narzędzia', 'Importuj', 'Ekspor', 'Online', and 'Pomoc'. Below the menu is a toolbar with icons for file operations, search, and various tools. The left sidebar shows a tree view with 'Konteksty' expanded, containing 'Domyslny kontekst' and 'Strony'. Under 'Strony', there is a selected item: 'https://2430-77-65-161-88.ngrok-free.app/'. The main content area features a large 'Welcome to ZAP' header and a message: 'ZAP jest łatwy w obsłudze, zintegrowanym narzędziem do testów penetracyjnych. Dzięki niemu można znaleźć luki w aplikacjach internetowych.' Below this are four buttons: 'Automated Scan' (blue lightning bolt), 'Manual Explore' (green lightning bolt), 'Support' (yellow lightning bolt), and 'Learn More' (blue question mark). At the bottom, there's a history tab, a search bar, and a progress bar showing '100%'. The bottom status bar displays 'Obecnie Skanowane: 0 Ilość Zadań: 6242 Nowe Alerty: 36 Eksport'.

The screenshot shows the ZAP interface with the following details:

**Network Request (Top):**

- Method: GET
- URL: <https://2430-77-65-161-88.ngrok-free.app>
- Status: HTTP/1.1 200 OK
- Headers:
  - Access-Control-Allow-Origin: \*
  - Content-Length: 2894
  - Content-Security-Policy: default-src 'self' https://cdn.ngrok.com 'unsafe-eval' 'unsafe-inline'; img-src data: w3.org/svg/2000
  - Content-Type: text/html
  - Referer-Policy: no-referrer
  - X-Content-Type-Options: nosniff
- Date: Tue, 28 Jan 2025 02:19:20 GMT

**Code Snippet (Middle):**

```
<!DOCTYPE html>
<html class="h-full" lang="en-US" dir="ltr">
  <head>
    <link rel="preload" href="https://cdn.ngrok.com/static/fonts/euclid-square/EuclidSquare-Regular-WebS.woff" as="font" type="font/woff" crossorigin="anonymous" />
    <link rel="preload" href="https://cdn.ngrok.com/static/fonts/euclid-square/EuclidSquare-RegularItalic-WebS.woff" as="font" type="font/woff" crossorigin="anonymous" />
    <link rel="preload" href="https://cdn.ngrok.com/static/fonts/euclid-square/EuclidSquare-Medium-WebS.woff" as="font" type="font/woff" crossorigin="anonymous" />
    <link rel="preload" href="https://cdn.ngrok.com/static/fonts/euclid-square/EuclidSquare-MediumItalic-WebS.woff" as="font" type="font/woff" crossorigin="anonymous" />
    <link rel="preload" href="https://cdn.ngrok.com/static/fonts/ibm-plex-mono/IBMPlexMono-Text-Woff" as="font" type="font/woff" crossorigin="anonymous" />
    <link rel="preload" href="https://cdn.ngrok.com/static/fonts/ibm-pex-mono/IBMPlexMono-TextItalic-Woff" as="font" type="font/woff" crossorigin="anonymous" />
    <link rel="preload" href="https://cdn.ngrok.com/static/fonts/ibm-pex-mono/IBMPlexMono-SemiBold.Woff" as="font" type="font/woff" crossorigin="anonymous" />
```

**Findings List (Bottom):**

- CSP: Wildcard Directive (6)**
  - CSP: script-src unsafe-inline (6)
  - CSP: style-src unsafe-inline (6)
  - Cross-Domain Misconfiguration (6)
  - Missing Anti-clickjacking Header (6)
  - Application Error Disclosure (4)
  - Cross-Domain JavaScript Source File Inclusion (6)
  - Information Disclosure - Debug Error Messages (4)
  - Server Leaks Information via "Powered-By" HTTP Response (1)
  - Strict-Transport-Security Header Not Set (18)
  - X-Content-Type-Options Header Missing (1)
  - Authentication Required Identified (1)

The screenshot shows the Sejsa Domyslna 2.15.0 application window. The top menu bar includes: Plk, Edytuj, Widok, Analizuj, Report, Narzdzia, Importuj, Eksport, Online, and Pomoc. Below the menu is a toolbar with various icons for file operations like Open, Save, Print, and Help.

The main workspace is divided into several panes:

- Left pane (Tree View):** Shows a hierarchical tree of findings:
  - Konteksty (Contexts)
  - Domyslny kontekst (Default context)
  - Strony (Pages)
  - Strona https://2430-77-65-161-88.ngrok-free.app
- Top Right pane (Details):** Displays the response headers for the selected page:

```
HTTP/1.1 200 OK
Access-Control-Allow-Origin: *
Content-Length: 898
Content-Security-Policy: default-src 'self' https://cdn.ngrok.com 'unsafe-eval' 'unsafe-inline'; img-src data: w3.org/svg/2000
Content-Type: text/html
Referer-Policy: no-referrer
X-Content-Type-Options: nosniff
Date: Tue, 28 Jan 2025 02:19:20 GMT
```
- Bottom Right pane (Code View):** Shows the HTML code of the page, highlighting several `<link rel="preload"` statements for various font files.
- Bottom Left pane (List View):** A detailed list of findings:
  - Zagrozenia (15)
    - CSP: Wildcard Directive (6)
    - CSP: script-src unsafe-inline (6)
    - CSP: style-src unsafe-inline (6)
  - Cross-Domain Misconfiguration (6)
    - Mising Anti-clickjacking Header (6)
    - Application Error Disclosure (4)
    - Cross-Domain JavaScript Source File Inclusion (6)
    - Information Disclosure - Debug Error Messages (4)
    - Server Leaks Information via "X-Powered-By" HTTP Response Header (2)
    - Strict-Transport-Security Header Not Set (18)
    - X-Content-Type-Options Header Missing (1)
    - Authentication Request Identified (1)
- Bottom Center pane (Summary):** A summary of the selected finding:

**Cross-Domain Misconfiguration**

URL: https://2430-77-65-161-88.ngrok-free.app/user

Ryziko: Medium

Zaufanie: Medium

Parametry: Atak

Dowód: Access-Control-Allow-Origin: \*

CWE ID: 264

WASC ID: 14

Zródło: Pasywne (10088 - Cross-Domain Misconfiguration)

Input Vector:

Opis:

Web browser data loading may be possible, due to a Cross Origin Resource Sharing (CORS) misconfiguration on the web server

The screenshot displays two separate ZAP sessions. The top session, titled 'Sesja Domyslna - ZAP 2.15.0', shows a successful 200 OK response from 'https://2430-77-65-161-88.ngrok-free.app'. The response details include:

- HTTP/1.1 200 OK
- Access-Control-Allow-Origin: \*
- Content-Length: 2084
- Content-Security-Policy: default-src 'self' https://cdn.ngrok.com 'unsafe-eval' 'unsafe-inline'; img-src data: w3.org/svg/2000
- Content-Type: text/html
- Referrer-Policy: no-referrer
- X-Content-Type-Options: nosniff
- Date: Tue, 28 Jan 2025 02:19:19 GMT

The page source code is shown, featuring multiple `<link rel="preload" href="https://cdn.ngrok.com/...>` statements for various font files.

The bottom session, also titled 'Sesja Domyslna - ZAP 2.15.0', shows an Internal Server Error (500) response from the same URL. The response details include:

- HTTP/1.1 500 Internal Server Error
- Content-Length: 52
- Content-Type: application/json; charset=utf-8
- Date: Tue, 28 Jan 2025 02:19:18 GMT
- Etag: W/34-n1Kccw1v/\NvB1Qk4oF1tDPro
- Ngrok-Agent-Ips: 77.65.161.88
- X-Powered-By: Express

The page source code contains the JSON object: `{"statusCode":500,"message":"Internal server error"}`.

In both sessions, the 'Zagrozenia' (Threats) tab is open, showing various security issues found in the application. The threats listed in the bottom session include:

- GSP Wildcard Directive (6)
- GSP script-src unsafeInline (6)
- GSP style-src unsafeInline (6)
- Cross-Domain Misconfiguration (6)
- Missing Anti-clickjacking Header (6)
  - Application Error Disclosure (4)
  - Cross-Domain JavaScript Source File Inclusion (6)
  - Information Disclosure - Debug Error Messages (4)
  - Server Leaks Information via "X-Powered-By" HTTP Response (1)
  - Strict-Transport-Security Header Not Set (18)
  - X-Content-Type-Options Header Missing (1)
  - Authentication Request Identified (1)

The screenshot shows the Sejsa Domyslna interface. At the top, there's a menu bar with Polish labels: Plik, Edytuj, Widok, Analizuj, Raport, Narzzedzia, Importuj, Eksport, Online, Pomoc. Below the menu is a toolbar with various icons. The main area has tabs for 'Strony' (websites) and 'Konteksty' (contexts). A context named 'Domyslny kontekst' is selected. Under 'Strony', a folder named 'https://2430-77-65-161-88.ngrok-free.app' is expanded. On the right, there's a detailed view of the page source code, showing various CSS and JavaScript imports. Below the browser view, a sidebar lists findings under 'Zagrozenia (15)':

- CSP: Wildcard Directive (6)
- CSP: script-src unsafe-inline (6)
- CSP: style-src unsafe-inline (6)
- Cross-Domain Misconfiguration (6)
- Missing Anti-clickjacking Header (6)
- Application Error Disclosure (4)
- Cross-Domain JavaScript Source File Inclusion (6)**
- Information Disclosure - Debug Error Messages (4)
- Server Leaks Information via "X-Powered-By" HTTP Response
- Strict-Transport-Security Header Not Set (18)
- X-Content-Type-Options Header Missing
- Authentication Request Identified

A specific item in the 'Cross-Domain JavaScript Source File Inclusion' section is highlighted in blue.

The screenshot shows the Sesja Domyslna - ZAP 2.15.0 interface. The top navigation bar includes: Plik, Edytuj, Widok, Analizuj, Raport, Narzędzia, Importuj, Eksport, Online, Pomoc. Below the menu is a toolbar with icons for file operations, search, and various tools. The main window has two panes. The left pane displays a tree view of 'Konteksty' (Contexts) and 'Strony' (Pages), with a selected item 'https://2430-77-65-161-88.ngrok-free.app'. The right pane shows a 'Raw View' of an HTTP response. The response details are as follows:

```
HTTP/1.1 500 Internal Server Error
Content-Length: 52
Content-Type: application/json; charset=utf-8
Date: Tue, 28 Jan 2025 02:19:18 GMT
Etag: W/"34-e1Kccu4s/VbA1Q4dF1tDfPro"
Nginx-Agent-Ips: 77.65.161.88
X-Powered-By: Express

{"statusCode":500,"message":"Internal server error"}
```

Below this, a 'Wynik' (Results) section is visible, showing a list of findings under 'Zugrożenia' (Vulnerabilities), including:

- CSP: Wildcard Directive (6)
- CSP: script-src unsafe-inline (6)
- CSP: style-src unsafe-inline (6)
- Cross-Domain Misconfiguration (6)
- Missing Anti-clickjacking Header (6)
- Application Error Disclosure (4)
- Cross-Domain JavaScript Source File Inclusion (6)

The bottom part of the interface shows a detailed view of an 'Information Disclosure - Debug Error Messages' finding, with fields like URL, Ryzyko (Risk), Zaufanie (Confidence), Parametr (Parameter), AIAk (AIAK), Dowód (Proof), CWE ID, WASC ID, Źródło (Source), Input Vector, and Opis (Description). The description notes: "The response appeared to contain common error messages returned by platforms such as ASP.NET, and Web-servers such as IIS and Apache. You can configure the list of common debug messages."

Sesja Domyslna - ZAP 2.15.0

Plik Edytuj Widok Analizuj Raport Narzdzia Importuj Eksport Online Pomoc

Tryb standardowy Szybki Start Zadanie Odpowiedz Requester

Konteksty Domyslny kontekst Strony https://2430-77-65-161-88.ngrok-free.app

Nagłówek: Raw View Ciało: Raw View

HTTP/1.1 400 Bad Request  
Content-Length: 50  
Content-Type: application/json; charset=utf-8  
Date: Tue, 28 Jan 2025 02:19:18 GMT  
Etag: W/"32-h73jH25KuzGwVgWf3G/KL1/o"  
Ngrok-Agent-Ips: 77.65.161.88  
X-Powered-By: Express

{"statusCode":400,"message":"Invalid credentials"}

Historia Szukaj Zagrozenia Wynik Progress Aktywne Skanowanie

Zagrożenia (15)  
CSP:Wildcard Directive (6)  
CSP:script-src unsafe-inline (6)  
CSP:style-src unsafe-inline (6)  
Cross-Domain Misconfiguration (6)  
Missing Anti-clickjacking Header (6)  
Application Error Disclosure (4)  
Cross-Domain JavaScript Source File Inclusion (6)  
Information Disclosure - Debug Error Messages (4)

Server Leaks Information via "X-Powered-By" HTTP Response (18)  
Strict-Transport-Security Header Not Set (18)  
X-Content-Type-Options Header Missing  
Authentication Request Identified

Obecnie skanowane 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Sesja Domyslna - ZAP 2.15.0

Plik Edytuj Widok Analizuj Raport Narzdzia Importuj Eksport Online Pomoc

Tryb standardowy Szybki Start Zadanie Odpowiedz Requester

Konteksty Domyslny kontekst Strony https://2430-77-65-161-88.ngrok-free.app

Nagłówek: Raw Request Ciało: Raw Request

HTTP/1.1 400 Bad Request  
Content-Length: 50  
Content-Type: application/json; charset=utf-8  
Date: Tue, 28 Jan 2025 02:19:18 GMT  
Etag: W/"32-h73jH25KuzGwVgWf3G/KL1/o"  
Ngrok-Agent-Ips: 77.65.161.88  
X-Powered-By: Express

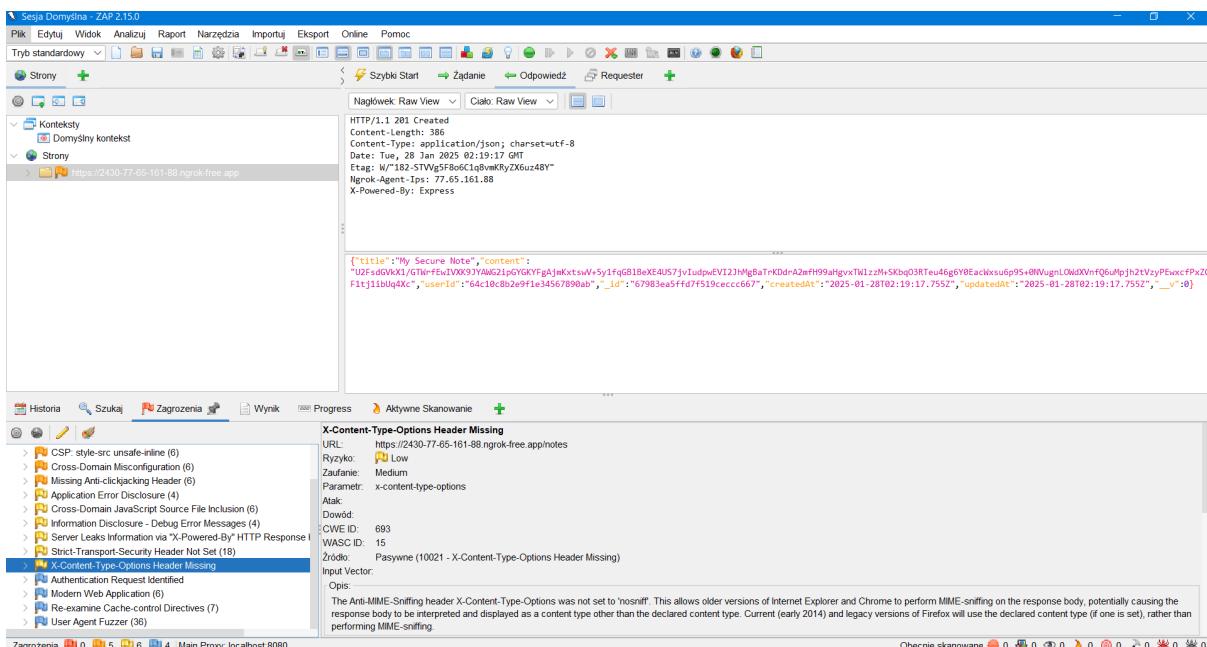
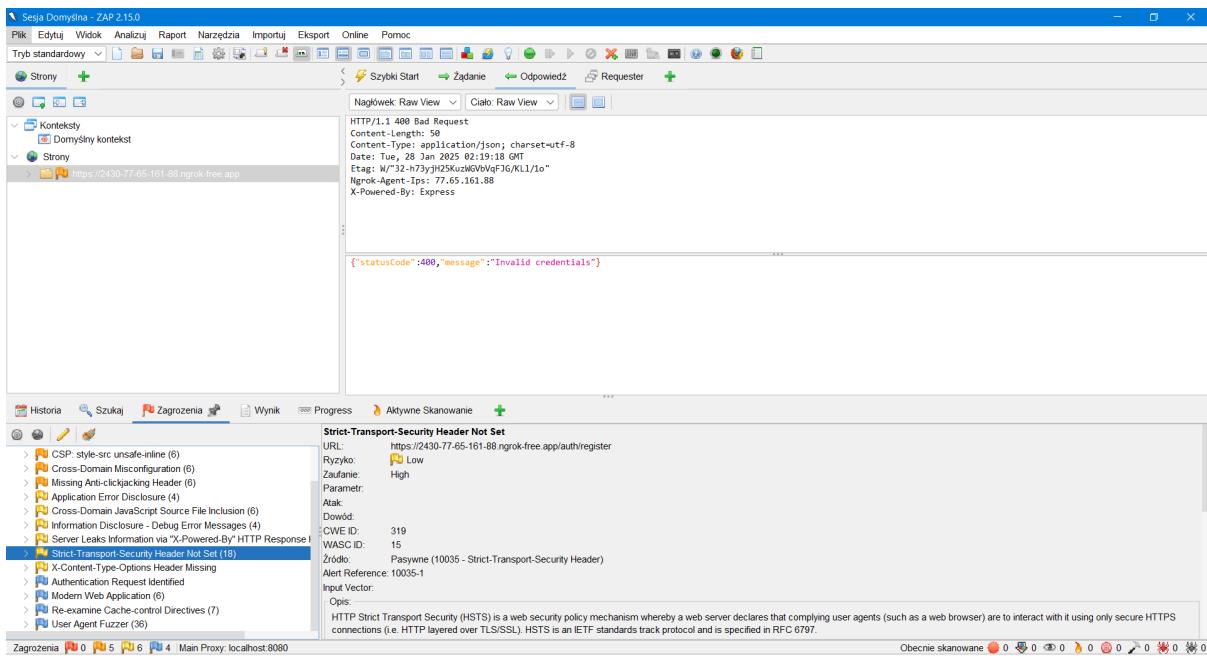
{"statusCode":400,"message":"Invalid credentials"}

Historia Szukaj Zagrozenia Wynik Progress Aktywne Skanowanie

Zagrożenia (15)  
CSP:style-src unsafe-inline (6)  
Cross-Domain Misconfiguration (6)  
Missing Anti-clickjacking Header (6)  
Application Error Disclosure (4)  
Cross-Domain JavaScript Source File Inclusion (6)  
Information Disclosure - Debug Error Messages (4)

Strict-Transport-Security Header Not Set (18)  
X-Content-Type-Options Header Missing  
Authentication Request Identified  
Modern Web Application (6)  
Re-examine Cache-control Directives (7)  
User Agent Fuzzer (36)

Obecnie skanowane 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0



The screenshot shows the ZAP Domyslna interface. The menu bar includes 'Plik', 'Edytuj', 'Widok', 'Analizuj', 'Raport', 'Narzędzia', 'Importuj', 'Ekspорт', 'Online', and 'Pomoc'. The toolbar has icons for file operations like Open, Save, Print, and a search bar. A top navigation bar includes 'Szybki Start', 'Zajdanie', 'Odpowiedz', and 'Requester'. The left sidebar shows a tree view with 'Strony' selected, expanded to show 'Konteksty' (with 'Domyslny kontekst') and 'Strony'. Under 'Strony', there's a node for 'https://2430-77-65-161-88.ngrok-free.app'. The main panel displays a request in the Requester tab. The 'Raw View' section shows a POST request to 'https://2430-77-65-161-88.ngrok-free.app/auth/login' with the following headers:

```
POST https://2430-77-65-161-88.ngrok-free.app/auth/login HTTP/1.1
host: 2430-77-65-161-88.ngrok-free.app
user-agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:125.0) Gecko/20100101 Firefox/125.0
pragma: no-cache
cache-control: no-cache
accept: application/json
content-type: application/json
content-length: 48
```

The 'Raw View' section also contains the JSON payload: `{"email": "zapproxy@example.com", "password": "ZAP"}`.

The screenshot shows the OWASP ZAP application's interface. The top navigation bar includes links for Historia, Szukaj, Zagrożenia, Wynik, Progress, and Aktywne Skanowanie. The main content area has a title "Authentication Request Identified". On the left, a tree view lists various security issues found during the scan, such as CSP: style-src unsafe-inline (6), Cross-Domain Misconfiguration (6), Missing Anti-clickjacking Header (6), Application Error Disclosure (4), Gross-Domain JavaScript File Source Disclosure (6), Information Disclosure - Debug Error Messages (4), Server Leaks Information via X-Powered-By HTTP Response (1), Strict-Transport-Security Header Not Set (18), and X-Content-Type-Options Header Missing. A specific item, "Authentication Request Identified", is highlighted with a blue selection bar at the bottom. The right side of the screen displays detailed information about this identified request, including the URL (https://2430-77-65-161-88.ngrok-free.app/auth/login), Risk Level (Informational), and a parameter named "email". It also lists the attack vector ("password") and provides links to CWE ID, WASC ID, and a source reference (Pasywne (10111 - Authentication Request Identified)). Below this, there is an "Other Info" section containing key-value pairs like "Input Vector" and "Opis". A note at the bottom states: "The given request has been identified as an authentication request. The 'Other Info' field contains a set of key-value lines which identify any relevant fields. If the request is in a context which has an Authentication Method set to 'Auto-Detect' then this rule will change the authentication to match the request identified." At the very bottom, the footer includes links for Zagrożenia, Main Proxy, localhost 8080, and a status bar showing "Obecnie skanowane" with various icons.

Sesja Domyslna - ZAP 2.15.0

Plik Edytuj Widok Analizuj Raport Narzdzia Importuj Online Pomoç

Tryb standardowy Szybki Start Zadanie Odpowiedz Requester

Strony + Nagłówki Raw View Ciało Raw View +

Konteksty Domyslny kontekst Strony https://2430-77-65-161-88.ngrok-free.app

HTTP/1.1 201 Created  
Content-Length: 386  
Content-Type: application/json; charset=utf-8  
Date: Tue, 28 Jan 2025 02:19:17 GMT  
Etag: W/182-STVG5t8eC1q8wKPyX6uz48Y"  
Ngrok-Agent-Ips: 77.65.161.88  
X-Powered-By: Express

{"title": "My Secure Note", "content": "U2FsdGVkX1/GTwFwVXK9jAYg2JpGqYKqFgAjmkxtswv5y1fqG81BxEx4U57jvIudpwEV12JHgBaTrKDrA2mfH99aflgvxTw1zzNsSkbgQ3RTeu4g6Y0Eackxsu6p95+0lVugnL0wdXvNfQ6aPjh2tVsyPEwxctFxZoFitJ1BdqX4c", "userId": "64:10:8b2e9f1a3456789ab", "id": "67983ea5ff7f519cccc667", "createdAt": "2025-01-28T02:19:17.752Z", "updatedAt": "2025-01-28T02:19:17.752Z", "v": 0}

Historia Szukaj Zagrozenia Wynik Progress Aktywne Skanowanie +

CSP style-src unsafe-inline (6)  
Cross-Domain Misconfiguration (6)  
Missing Anti-clickjacking Header (6)  
Application Error Disclosure (4)  
Cross-Domain JavaScript File Inclusion (6)  
Information Disclosure - Debug Error Messages (4)  
Server Leaks Information via "X-Powered-By" HTTP Response (1)  
Strict-Transport-Security Header Not Set (18)  
X-Content-Type-Options Header Missing (1)  
Authentication Request Identified (1)  
Modern Web Application (6)  
Re-examine Cache-control Directives (7)  
User Agent Fuzzer (36)

Re-examine Cache-control Directives

URL: https://2430-77-65-161-88.ngrok-free.app/notes  
Rzyko: Informalny  
Zaufanie: Low  
Parametr: cache-control  
Atak: Dowód:  
CWE ID: 525  
WASC ID: 13  
Zródło: Pasywne (10015 - Re-examine Cache-control Directives)  
Input Vector:  
Opis:  
The cache-control header has not been set properly or is missing, allowing the browser and proxies to cache content. For static assets like css, js, or image files this might be intended, however, the resources should be reviewed to ensure that no sensitive content will be cached.

The screenshot shows the Seja Domyslna application interface. The top menu bar includes 'Plik', 'Edytuj', 'Widok', 'Analizuj', 'Report', 'Narzędzia', 'Importuj', 'Ekspорт', 'Online', 'Pomoc'. Below the menu is a toolbar with various icons. The main window has a left sidebar with 'Konteksty' and 'Strony' sections, and a central pane displaying a note from 'https://2430-77-65-161-88.ngrok-free.app'. The note content is as follows:

```
HTTP/1.1 201 Created
Content-Length: 36
Content-Type: application/json; charset=utf-8
Date: Tue, 28 Jan 2025 02:21:04 GMT
Etag: W/182-jeodxb00/cpcT9keeo+xEKE1Vg"
Ngrok-Agent-Ips: 77.65.161.88
X-Request-Id: 02d9f333-43c4-43e0-8330-000000000000

{
    "title": "My Secure Note",
    "content": "U2ZfGvKXL1IIPxhJWxhBz9MKhhs0la0nU9CjFrjNjzkJuR8hsAr0E2qui64cuyxGPE2KV1F6efBHAmau1stFe1hInIq0zLoA2vGab1FFrvilZnEa98D0k1joiUnkkLBNTNxurOeCcoQV1122jnzd2n6po1+A3c7Y8u0UzhQtPwFsC",
    "userId": "64c10c8b2e9fe3456789ab",
    "id": "67983f10ff47f519ecccbc",
    "createdAt": "2025-01-28T02:21:04.312Z",
    "updatedAt": "2025-01-28T02:21:04.312Z"
}
```

The bottom section of the interface shows a list of findings under 'User Agent Fuzzer'.

Kategoria	Opis
CSP: style-src unsafe-inline (6)	
Cross-Domain Misconfiguration (6)	
Missing Anti-clickjacking Header (6)	
Application Error Disclosure (4)	
Cross-Domain JavaScript Source File Inclusion (6)	
Information Disclosure - Debug Error Messages (4)	
User Leak Information via X-Powered-By HTTP Response (1)	
X-Content-Security-Header Not Set (18)	
X-Content-type-Options Header Missing	
Authentication Request Identified	
Modern Web Application (6)	
Re-examine Cache-control Directives (7)	
User Agent Fuzzer (36)	

## Testy aplikacji wykonane przy pomocy programu MobSF:

The screenshot shows the MobSF web application's login interface. At the top, there is a navigation bar with links for RECENT SCANS, STATIC ANALYZER, DYNAMIC ANALYZER, API, DONATE, DOCS, ABOUT, and a search bar. Below the navigation bar is a "Sign in to access" message followed by the MobSF logo and a login form. The login form contains fields for "mobsf" and a password, and a "Sign In" button. At the bottom of the page, there is a footer with copyright information: "© 2025 Mobile Security Framework - MobSF | Ajin Abraham | OpenSecurity." and "Version 4.3.0".

The screenshot shows the main analysis results page of the MobSF web application. On the left, there is a sidebar with various analysis options: Static Analyzer, Information, Scan Options, Signer Certificate, Permissions, Android API, Browsable Activities, Security Analysis, Malware Analysis, Reconnaissance, Components, PDF Report, Print Report, and Start Dynamic Analysis. The main content area is divided into several sections: APP SCORES (Security Score: 38/100, Trackers Detection: 0/432), FILE INFORMATION (File Name: application-f8b61031-74f2-4319-9269-c237fc35913.apk, Size: 162.31MB, MD5: 9f149e3059c063ab58ee312de0561d04, SHA1: e7zf0f145fb118044d7d121e1f2ee972de7d6307, SHA256: 6799888ed8482f221a145268832f92ed24cd421b9800fdf880ac3726c17b958), APP INFORMATION (App Name: frontend-aplikacja, Package Name: com.czeczu7.frontendaplikacja, Main Activity: com.czeczu7.frontendaplikacja.MainActivity, Target SDK: 34 [Min SDK: 24, Max SDK: 34], Android Version Name: 1.0.0, Android Version Code: 4), and four summary cards: EXPORTED ACTIVITIES (2/5, View All), EXPORTED SERVICES (0/0, View All), EXPORTED RECEIVERS (1/1, View All), and EXPORTED PROVIDERS (0/3, View All). Below these cards are sections for SCAN OPTIONS (Rescan, Manage Suppressions, Start Dynamic Analysis, Scan Logs) and DECOMPILED CODE (View AndroidManifest.xml, View Source, View Smali, Download Java Code, Download Small Code, Download APK). There is also a SIGNER CERTIFICATE section indicating that the binary is signed.

APPLICATION PERMISSIONS				
PERMISSION	STATUS	INFO	DESCRIPTION	CODE MAPPINGS
android.permission.INTERNET	normal	full Internet access	Allows an application to create network sockets.	<a href="#">Show Files</a>
android.permission.READ_EXTERNAL_STORAGE	dangerous	read external storage contents	Allows an application to read from external storage.	<a href="#">Show Files</a>
android.permission.SYSTEM_ALERT_WINDOW	dangerous	display system-level alerts	Allows an application to show system-alert windows. Malicious applications can take over the entire screen of the phone.	<a href="#">Show Files</a>
android.permission.USE_BIOMETRIC	normal	allows use of device-supported biometric modalities.	Allows an app to use device supported biometric modalities.	<a href="#">Show Files</a>
android.permission.USE_FINGERPRINT	normal	allow use of fingerprint	This constant was deprecated in API level 28. Applications should request USE_BIOMETRIC instead.	
android.permission.VIBRATE	normal	control vibrator	Allows the application to control the vibrator.	<a href="#">Show Files</a>
android.permission.WRITE_EXTERNAL_STORAGE	dangerous	read/modify/delete external storage contents	Allows an application to write to external storage.	<a href="#">Show Files</a>
com.czechu7.frontendaplikacja.DYNAMIC_RECEIVER_NOT_EXPORTED_PERMISSION	unknown	Unknown permission	Unknown permission from android reference	

Showing 1 to 8 of 8 entries

Previous 1 Next

MANIFEST ANALYSIS				
HIGH		WARNING	INFO	SUPPRESSED
NO	ISSUE	SEVERITY	DESCRIPTION	OPTIONS
1	App can be installed on a vulnerable uppatched Android version [minSdk=24]	high	This application can be installed on an older version of android that has multiple unfixed vulnerabilities. These devices won't receive reasonable security updates from Google. Support an Android version => 10, API 29 to receive reasonable security updates.	<a href="#">🔗</a>
2	Clear text traffic is Enabled For App [android:usesCleartextTraffic=true]	high	The app intends to use cleartext network traffic, such as cleartext HTTP, FTP stacks, DownloadManager, and MediaPlayer. The default value for apps that target API level 27 or lower is "true". Apps that target API level 28 or higher default to "false". The key reason for avoiding cleartext traffic is the lack of confidentiality, authenticity, and protections against tampering; a network attacker can eavesdrop on transmitted data and also modify it without being detected.	<a href="#">🔗</a>
3	Debug Enabled For App [android:debuggable=true]	high	Debugging was enabled on the app which makes it easier for reverse engineers to hook a debugger to it. This allows dumping a stack trace and accessing debugging helper classes.	<a href="#">🔗</a>
4	Application Data can Be backed up [android:allowBackup=true]	warning	This flag allows anyone to backup your application data via adb. It allows users who have enabled USB debugging to copy application data off of the device.	<a href="#">🔗</a>
5	<b>Activity</b> (expo.modules.devlauncher.launcher.DevLauncherActivity) is not Protected. [android:exported=true]	warning	An Activity is found to be shared with other apps on the device therefore leaving it accessible to any other application on the device.	<a href="#">🔗</a>
6	<b>Activity</b> (expo.modules.devmenu.DevMenuActivity) is not Protected. [android:exported=true]	warning	An Activity is found to be shared with other apps on the device therefore leaving it accessible to any other application on the device.	<a href="#">🔗</a>
7	<b>Broadcast Receiver</b> (androidx.profileinstaller.ProfileInstallReceiver) is Protected by a permission, but the protection level of the permission should be checked.	warning	A Broadcast Receiver is found to be shared with other apps on the device therefore leaving it accessible to any other application on the device. It is protected by a permission which is not defined in the analysed application. As a result, the protection level of the permission should be checked where it is defined. If it is set to normal or	<a href="#">🔗</a>

Code Analysis					
HIGH 2	WARNING 4	INFO 2	SECURE 0	SUPPRESSED 0	
<span style="font-size: small;">Search: <input type="text"/></span>					
NO	ISSUE	SEVERITY	STANDARDS	FILES	OPTIONS
1	The App logs information. Sensitive information should never be logged.	info	<b>CWE:</b> CWE-532: Insertion of Sensitive Information into Log File <b>OWASP Top 10:</b> M5: Insufficient Cryptography <b>OWASP MASVS:</b> MSTG-STORAGE-3	<a href="#">Show Files</a>	
2	MD5 is a weak hash known to have hash collisions.	warning	<b>CWE:</b> CWE-327: Use of a Broken or Risky Cryptographic Algorithm <b>OWASP Top 10:</b> M5: Insufficient Cryptography <b>OWASP MASVS:</b> MSTG-CRYPTO-4	<a href="#">expo/modules/asset/AssetModule.java</a> <a href="#">expo/modules/filesystem/FileSystemModule.java</a> <a href="#">expo/modules/filesystem/react/FileSystemFile.java</a>	
3	Debug configuration enabled. Production builds must not be debuggable.	high	<b>CWE:</b> CWE-919: Weaknesses in Mobile Applications <b>OWASP Top 10:</b> M1: Improper Platform Usage <b>OWASP MASVS:</b> MSTG-RESILIENCE-2	<a href="#">Show Files</a>	
4	Files may contain hardcoded sensitive information like usernames, passwords, keys etc.	warning	<b>CWE:</b> CWE-312: Cleartext Storage of Sensitive Information <b>OWASP Top 10:</b> M9: Reverse Engineering <b>OWASP MASVS:</b> MSTG-STORAGE-14	<a href="#">Show Files</a>	
5	Remote WebView debugging is enabled.	high	<b>CWE:</b> CWE-919: Weaknesses in Mobile Applications <b>OWASP Top 10:</b> M1: Improper Platform Usage <b>OWASP MASVS:</b> MSTG-RESILIENCE-2	<a href="#">com/reactnativecommunity/webview/ITNCWebViewManagerImpl.java</a>	

## 3. Wnioski

Projekt "Aplikacja do przechowywania wrażliwych notatek z zabezpieczeniem przed atakiem Man-in-the-Middle" pozwolił na stworzenie nowoczesnego i bezpiecznego rozwiązania do zarządzania poufnymi informacjami. Wdrożone mechanizmy zabezpieczeń skutecznie chronią dane użytkownika zarówno przed atakami zewnętrznymi, jak i nieautoryzowanym dostępem do urządzenia.

Wnioski:

- Implementacja lokalnego szyfrowania danych znacząco podnosi poziom bezpieczeństwa aplikacji.
- Ograniczenie dostępu offline i funkcja zdalnego usuwania danych zwiększa ochronę w przypadku utraty urządzenia.

Projekt pokazał, że możliwe jest stworzenie bezpiecznej aplikacji przy użyciu technologii takich jak React Native, NestJS i MongoDB. Umożliwił również praktyczne zapoznanie się z zagadnieniami bezpieczeństwa w aplikacjach mobilnych.