

Лабораторная работа № 11

Тема: Firebase Authentication - Безопасность и авторизация.

Цель работы: Настроить систему регистрации и авторизации пользователей. Создать личный кабинет с базовой информацией о пользователе. Настроить правила безопасности для защиты данных.



Обязательные требования для всех вариантов:

- Настроить Authentication в Firebase Console
- Реализовать регистрацию пользователей (email/пароль или другой вариант)
- Реализовать вход в систему
- Реализовать выход из системы
- Настроить отслеживание состояния авторизации
- Создать личный кабинет пользователя
- Настроить базовые правила безопасности Firestore

Вариант 1: "Библиотека".

Задание: Добавить систему регистрации и авторизации в проект "Библиотека книг". Создать личный кабинет читателя.

Этапы реализации:

Шаг 1: Настройка Firebase.

В Firebase Console:

- Включите Authentication → "Начать"
- Выберите "Email/Пароль" (или другой) в методах входа
- Нажмите "Сохранить"

Правила безопасности Firestore (пример):

```
rules_version = '2';
service cloud.firestore {
  match /databases/{database}/documents {
    // 1. СНАЧАЛА ВСЁ ЗАПРЕЩАЕМ
    match /{document=**} {
      allow read, write: if false;
    }
  }
}
```

```
// 2. Теперь разрешаем по-немногу:
```

```
// 2.1. Книги могут читать ВСЕ (даже неавторизованные)
```

```
match /books/{bookId} {  
    allow read: if true; // Чтение всем  
    allow write: if false; // Пока никто не может писать  
}
```

```
// 2.2. Профили пользователей - только свои
```

```
match /users/{userId} {  
    allow read, write: if  
        // Должен быть авторизован  
        request.auth != null &&  
        // И может работать только со своим профилем  
        request.auth.uid == userId;  
    }  
}  
}
```

Шаг 2: Интерфейс авторизации.

Создать шапку с состоянием:

```
html  
<header>  
    <h1> Электронная библиотека</h1>  
    <div id="auth-panel">  
        <!-- Для гостя: -->  
        <button id="login-btn">Войти</button>  
        <button id="register-btn">Зарегистрироваться</button>  
  
        <!-- Для авторизованного: -->  
        <span id="welcome-text" class="hidden">Привет, </span>  
        <button id="logout-btn" class="hidden">Выйти</button>  
    </div>  
</header>
```

Форма регистрации (простейшая):

- Email (обязательно)
- Пароль (минимум 6 символов)
- Кнопка "Зарегистрироваться"

Форма входа:

- Email
- Пароль
- Кнопка "Войти"

Шаг 3: Работа с данными пользователя.

При регистрации сохранять в Firestore:

```
// Коллекция "users", документ = userId
{
  email: "user@example.com",
  displayName: "", // можно пустое
  createdAt: "дата регистрации",
  role: "reader" // просто текст
}
```

В личном кабинете показывать:

- Email пользователя
- Дата регистрации
- Простая надпись: "Вы вошли как читатель"
- Обновлять шапку при входе/выходе

Шаг 4: Проверка работы.

- Зарегистрировать нового пользователя
- Войти под этими данными
- Обновить страницу - пользователь должен остаться
- Выйти и войти под другим пользователем

Вариант 2: "Кинотеатр".

Задание: Добавить систему регистрации и авторизации в проект "Кинотеатр". Создать личный профиль зрителя.

Этапы реализации:

Шаг 1: Настройка Firebase.

В Firebase Console:

- Активируйте Authentication
- Метод входа: Email/Password " (или другой)
- Настройте домены авторизации (оставьте по умолчанию)

Правила безопасности (Пример):

```

rules_version = '2';
service cloud.firestore {
    match /databases/{database}/documents {
        // ПРАВИЛО 1: ВСЁ ЗАПРЕЩЕНО ПО УМОЛЧАНИЮ
        match /{document=**} {
            allow read, write: if false;
        }

        // ПРАВИЛО 2: Фильмы доступны всем для чтения
        match /movies/{movield} {
            allow read: if true; // Читать могут все
            allow write: if false; // Писать пока никто не может
        }

        // ПРАВИЛО 3: Профили - только свои
        match /profiles/{userId} {
            allow read: if
                // Читать профиль могут:
                // 1. Авторизованные пользователи
                request.auth != null &&
                // 2. Только свой профиль
                request.auth.uid == userId;

            allow write: if
                // Писать в профиль могут:
                // 1. Авторизованные пользователи
                request.auth != null &&
                // 2. Только свой профиль
                request.auth.uid == userId;
        }
    }
}

```

Шаг 2: Формы авторизации.

Кнопки в шапке сайта:

```

html
<div id="user-controls">
    <!-- Гость видит: -->
    <button onclick="showLogin()">Вход</button>
    <button onclick="showRegister()">Регистрация</button>

```

```
<!-- Авторизованный видит: -->
<span id="user-greeting" class="hidden"></span>
<button onclick="logout()" class="hidden">Выйти</button>
</div>
```

Форма регистрации (всплывающее окно или блок):

- Email
- Пароль
- Имя зрителя (необязательно)

Форма входа:

- Email
- Пароль
- Кнопка "Войти"

Шаг 3: Профиль пользователя.

Создать коллекцию profiles в Firestore:

// При регистрации сохранять:

```
{
  email: "user@example.com",
  fullName: "", // если ввели
  avatar: "", // можно пусто
  memberSince: "дата регистрации",
  favoriteGenre: "" // можно пусто
}
```

Страница или блок "Мой профиль":

- Показывать только авторизованным
- Выводить данные из Firestore
- Простая форма для изменения имени

Приветствие в шапке:

- "Привет, [Имя]!" или "Привет, [Email]!"

Шаг 4: Тестирование.

- Проверить регистрацию нового зрителя
- Проверить вход в систему
- Проверить сохранение состояния (обновить страницу)
- Проверить выход из системы

 Отчет должен содержать (см. образец):

- номер и тему лабораторной работы;
- фамилию, номер группы студента и вариант задания;
- Скриншот Firebase Console с вашим проектом
- Скриншот Firestore с данными
- HTML файл с кодом
- Скриншот работающей страницы в браузере
- пояснения, если необходимо;
- выводы.

Отчеты в формате pdf отправлять на email:

colledge20education23@gmail.com