

## **Задание: Студенты и их оценки.**

**Цель:** Научиться моделировать сущности с помощью классов, хранить их в Firebase и отображать в интерфейсе.

### **Навыки:**

- создание классов и приватных полей;
- использование геттеров/сеттеров;
- работа с DOM (createElement, appendChild);
- интеграция с Firebase (Firestore или Realtime Database);
- сериализация объектов через toJSON().

### **Часть 1. Базовый функционал.**

Класс Student должен содержать:

- Приватные поля:
  - #name — имя студента;
  - #grades — массив оценок.
- Публичные методы:
  - constructor(name, grades = []) — принимает имя и массив оценок;
  - addGrade(grade) — добавляет оценку в массив;
  - getAverage() — возвращает среднюю оценку;
  - toJSON() — возвращает «чистый объект» для сохранения в Firebase;
  - render(parent) — отображает карточку студента в DOM (имя + оценки + средняя оценка).

### **Часть 2. Интерфейс страницы.**

В HTML должны быть:

- форма <form> с <input> для имени студента и кнопкой «Добавить»;
- контейнер <div id="students"></div> для списка студентов.

### **Часть 3. Firebase.**

- Подключить Firebase SDK.
- При добавлении студента:

- создать экземпляр класса Student;
- сохранить его в Firebase (addDoc или push);
- отрендерить в DOM.
- При загрузке страницы — загрузить всех студентов из Firebase и отобразить их на странице.

#### Часть 4. Расширенные задания.

##### Добавление оценок:

- у каждой карточки сделать кнопку «Добавить оценку»;
- при клике — ввод числа через prompt;
- вызвать метод addGrade(), пересчитать среднее, обновить Firebase.

##### Редактирование имени:

- сделать сеттер set name(value), который меняет имя студента;
- добавить кнопку «Редактировать» в карточке.

##### Удаление студентов:

- кнопка «Удалить» в карточке удаляет запись из Firebase и DOM.

##### Реактивность:

- использовать onSnapshot (Firestore) или onValue (Realtime DB), чтобы студенты обновлялись на странице автоматически при изменениях в БД.



Пример данных в Firebase

```
{
  "students": {
    "id1": {
      "name": "Иван Петров",
      "grades": [5, 4, 3]
    },
    "id2": {
      "name": "Мария Иванова",
      "grades": [5, 5, 4]
    }
  }
}
```

```
}  
}  
}
```

#### 💡 Итоговое поведение

- При вводе имени в форму создаётся новый студент (с пустым списком оценок).
- Студент сохраняется в Firebase и появляется на странице.
- Можно добавить ему оценки → они сохраняются в Firebase и сразу отображаются.
- Средняя оценка считается автоматически.
- Можно редактировать имя и удалять студента.