ЛАБОРАТОРНАЯ «МАРШРУТИРИЗАЦИЯ».

**КОМПОНЕНТ APP**

var colorContext = createContext<List<Color>>()  
val shortOutput = createContext("Full")

Созданы 2 контекста, для того чтобы их передавать с помощью провайдера в другие компоненты.

val courses = courseList.map {useState(it)} – список состояний, где каждое состояние является курсом.

CMosePicker – компонент для выбора режима.

СColor – компонент для окрашивания оценок.

colorContext.Provider(ListColor) и shortOutput.Provider(mode) – провайдеры контекста, которые предоставляют информацию о цвете и оценке другим компонентам.

**HashRouter** – маршрутизатор. Компонент, который является провайдером информации и маршрутах. Все компоненты, которые внутри могут узнавать какой сейчас маршрут и в соответствии с этим прорисовываться (узнаем у него какой курс или какой список успеваемости).

courses.map – создаем ссылку для каждого курса

progress.map – создаем ссылку для каждого списка успеваемости студентов

**Link** – компонент для создания ссылки.

Имеет **2** аргумента:

+name – что написать (на ссылку, на которую будет нажимать пользователь)

to = name – куда перейти при нажатии

**Routes** – компонент, который говорит, что внутри мы прописываем компоненты, которые будут меняться в зависимости от маршрута.  
**Route** – как прорисовывать каждый маршрут

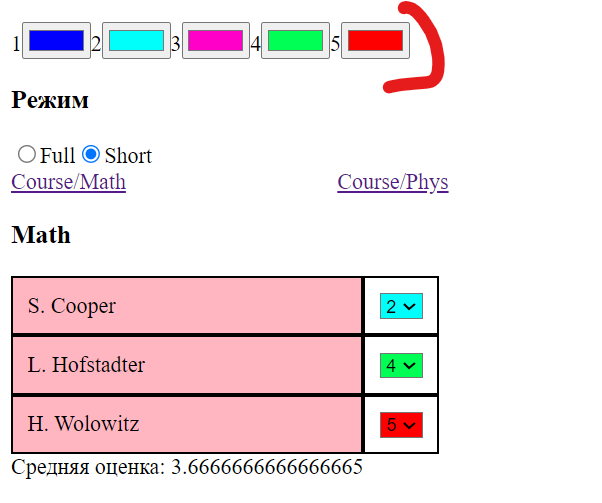
У него 2 аргумента:

Path – путь (то что после решетки в адресной строке).

Element – передаем готовый элемент (создаем его из компонента из функции create).

**Компонент CColor**

Компонент, предназначенный для отображения цветов, которые будут выбираться для каждой возможной оценки.



**Компонент CStudentList**

Предназначен для вывода таблицы с двумя колонками: студент и оценка. Пользователь может изменять оценки студентам (с помощью выпадающего списка, тег select). Внизу таблицы выводится средняя оценка.

**Компонент CGrade**

Используется для выставления оценки

Для элемента ввода используется функция onChange, которая берет объект из события (it.target) и вызывает функцию setGrade.

Цвет передается через контекст с помощью котекста useContext.

**Компонент CCourseRicker**

Анализирует параметры маршрута и в зависимости от этого отображает нужный курс.

**Компонент СGradePicker**

Анализирует параметры маршрута и в зависимости от этого отображает нужный список успеваемости студентов (отличники/хорошисты/двоечники).