

## 20. Учебный проект: меняй-удаляй (Часть 1)

### Задание

#### Создать модель данных

1. `src/models/tasks.js`,  
2. `2` .  
3. `id` : `id` ( `id` , `BoardController` .  
4. `main.js` ,  
5. `main.js` `BoardController` .  
6. `BoardController` .

#### Фильтрация

1. — ,  
 . render.  
 main.js render .  
 2. — ,  
 .  
 3. BoardController ?  
 !

1.  
 2. ,  
 3. BoardController ,  
 4. ( :  
 )  
 1.  
 2. ,  
 .

## Удаление и добавление задач

.  
 .  
 ( — ) ,  
 — .  
 1. onDataChange  
 null, null. :  
 ( ) .

2.

click

,

onDataChange

null.

3.

,

null

onDataChange

,

null

4.

,

1 1,

5.

,

,

(

),

,

## Выполненные пункты ТЗ

- «Add new task».

,

- .

- :

- —

;

- ( )

«

»;

- —

«Save»

- :

:

- 1 ;
- 140 ;
- .
- .
- , ,
- , , .
- .
- «Save». , .
- :
- All — .
- Overdue — .
- Today — .
- Favorites — .
- Repeating — .
- Tags — .
- Archive — .
- , .
- — .
- / / / / .

- `disabled`.
- .
- ( )  
«Archive».

## 21. Личный проект: меняй-удаляй (Часть 1)

### Большое путешествие

#### Задание

#### Создать модель данных

1. `src/models/points.js`,
2. 2 . ,
3. 2 : `id` — ( `id` , `TripController`.
4. `main.js` ,

5. `main.js` `TripController` `render`.

6. `TripController`  
—  
.

## Фильтрация

1. — `render`.  
`main.js` `render`.

2. —  
,

3. `TripController` ?  
!

1.  
.

2. ,  
.

3. `TripController` ,  
.

## Удаление и добавление точек маршрута

—  
( — )  
,  
.

1. `onDataChange`  
`null.`  
`null,`  
`:`  
`(`  
`)`  
`.`
2. `click`  
`,`  
`null.`  
`onDataChange`
3. `null`  
`,`  
`onDataChange`  
`—`  
`,`  
`null`  
`.`
4. `,`  
`.`  
`1 1,`  
`,`  
`.`
5. `.`  
`,`  
`,`  
`,`  
`.`

## Выполненные пункты ТЗ

- «New Event».  
`,`  
`(`  
`)`.
- «New Event».  
`.`
- «New Event»  
`,`  
`.`
- `:`
- «Save».  
`.`

- «Cancel».
- 
- 
- 
- 
- «Save».
- «Delete».
- 
- 
- 
- Everything —
- Future —
- Past —
- 

## Киноман

## Задание



## Создать модель данных

1. `src/models/movies.js`,  
2. `2` .  
3. `id` : `id` ( `id` , `PageController` .  
4. `main.js` ,  
5. `main.js` `PageController` .  
6. `PageController` —

## Фильтрация

1. `render` .  
2. `main.js` `render` .

3. `PageController` ?  
!

1.

2. ,

3. `PageController` ,

4. ( :  
)

1.

2. ,

## Удаление и добавление комментариев

(  
)  
,  
:  
:

1. `onDataChange`  
`null`. :  
`null`, ( )

2. `click`  
`onDataChange`  
`null`.

3. `null` `onDataChange` ,

• , `null`  
•  
4.  
( !).  
( ),  
( ).

## Выполненные пункты ТЗ

- 140 , 139  
(...).
- 
- :
- ☐ ;
- ☐ ;
- ☐ ;
- ☐ ;
- ☐ .
- `YYYY/MM/DD HH:MM` .  
: `2019/12/31 23:59` .
- ( : `sleeping`, `neutral-face` `grinning` ).  
 ,  
 .
-

• Ctrl/Command +  
Enter.

• «Delete», .

• :

○ «All movies» — ;

○ «Watchlist» — , (Watchlist);

○ «History» — (Already watched);

○ «Favorites» — , (Favorites).

• , «All movies» .

• , .

• ,  
: «Show more». (

5). , «Show more»

1. `markup`.  
 , `visually-hidden`, уже .
2. `main.js` .  
 , .
3. `AbstractComponent` `show/hide` .  
`visually-hidden` .
4. `BoardController` ,  
`visually-hidden` .
5. , `main.js` — ,  
 , — .

## Безжалостная статистика

- `chart.js` .  
 — ,
1. `npm` `chart.js` `chartjs-plugin-`  
`datalabels` .
2. `chart.js` `chartjs-plugin-datalabels` ,  
`import` .
3. `flatpickr` .  
 , ,  
 . `flatpickr`  
 (<https://flatpickr.js.org/examples/#range-calendar>).  
 4. .

5. `chart.js` : Done by: `Tags` Done by: Colors.  
chart.js — <https://www.chartjs.org/docs/latest/>.

6. .

7. .

## Выполненные пункты ТЗ

- : « » « ».
- «Tasks» .
- «Archive» , «Archive». 1.4  
« ».
- «Statistic» .
- ( « » )  
: « » (Done by: Tags) «  
» (Done by: Colors).
- 15 , `09.03-15.03`.
- « » `chart.js`.
- .
- `flatpickr`.

## 23. Личный проект: меняй-удаляй (Часть 2)

# Задание

Вам предстоит реализовать приложение, которое будет отображать статистику безжалостности различных стран. Приложение будет состоять из нескольких экранов, которые будут отображаться в зависимости от того, на каком этапе находится пользователь.

## Большое путешествие

### Смена экранов

Вам предстоит реализовать приложение, которое будет отображать статистику безжалостности различных стран. Приложение будет состоять из нескольких экранов, которые будут отображаться в зависимости от того, на каком этапе находится пользователь.

1. В файле `markup` необходимо реализовать разметку для экрана. В файле `main.js` необходимо реализовать логику, которая будет управлять сменой экранов. В файле `visually-hidden` необходимо реализовать компонент, который будет использоваться для скрытия элементов, которые не должны быть видны пользователю.
2. В файле `main.js` необходимо реализовать логику, которая будет управлять сменой экранов. В файле `visually-hidden` необходимо реализовать компонент, который будет использоваться для скрытия элементов, которые не должны быть видны пользователю.
3. В файле `AbstractComponent` необходимо реализовать абстрактный класс, который будет использоваться для создания компонентов. В файле `show/hide` необходимо реализовать функции, которые будут использоваться для показа и скрытия элементов. В файле `visually-hidden` необходимо реализовать компонент, который будет использоваться для скрытия элементов, которые не должны быть видны пользователю.
4. В файле `TripController` необходимо реализовать класс, который будет использоваться для управления поездкой. В файле `visually-hidden` необходимо реализовать компонент, который будет использоваться для скрытия элементов, которые не должны быть видны пользователю.
5. В файле `main.js` необходимо реализовать логику, которая будет управлять сменой экранов. В файле `visually-hidden` необходимо реализовать компонент, который будет использоваться для скрытия элементов, которые не должны быть видны пользователю.

### Безжалостная статистика

Вам предстоит реализовать приложение, которое будет отображать статистику безжалостности различных стран. Приложение будет состоять из нескольких экранов, которые будут отображаться в зависимости от того, на каком этапе находится пользователь. В файле `chart.js` необходимо реализовать компонент, который будет использоваться для отображения статистики.

1. npm `chart.js` `chartjs-plugin-`  
`datalabels`.

2. `chart.js` `chartjs-plugin-datalabels`,  
`import`.

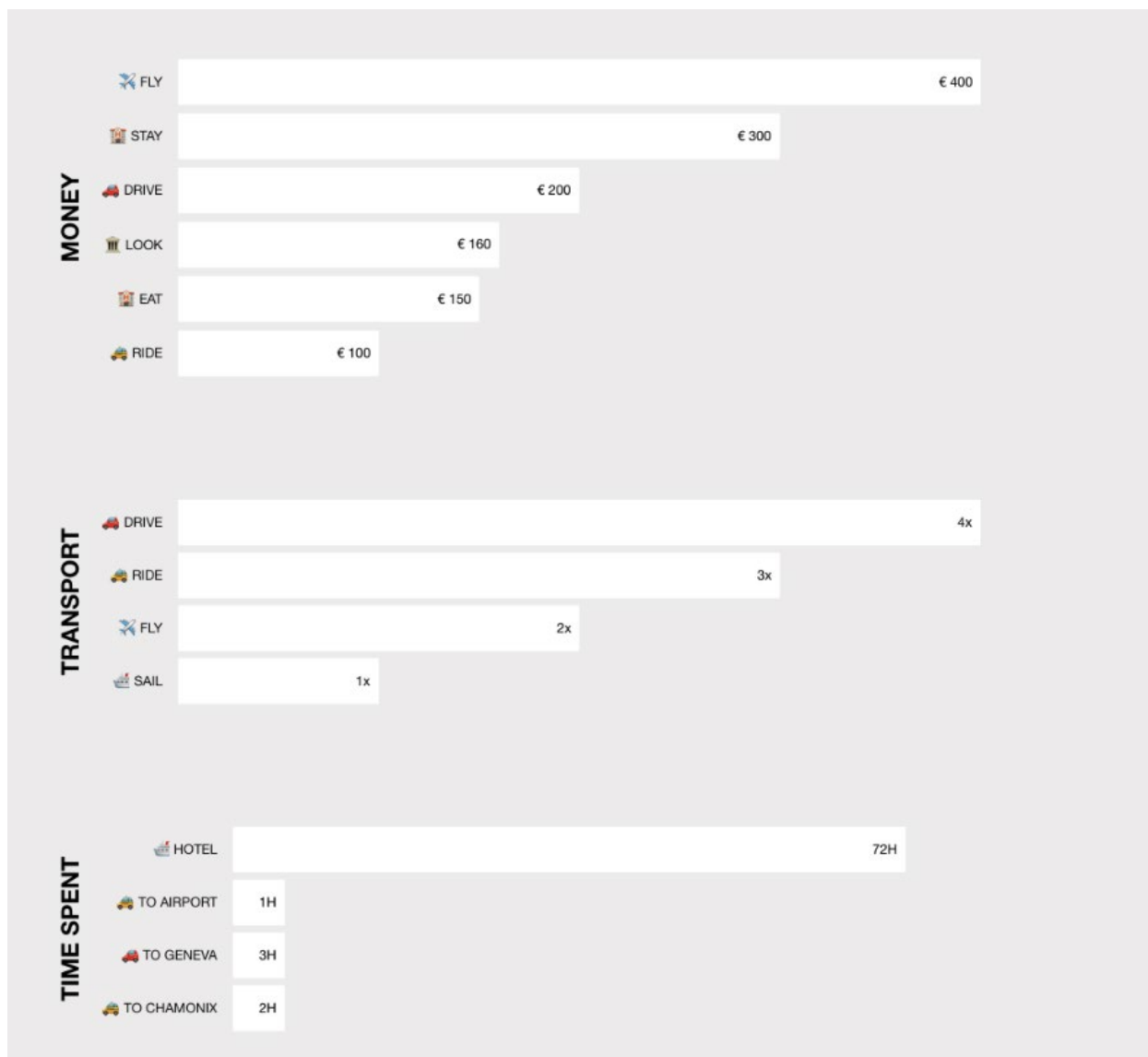
3. `chart.js` : `Money`, `Transport` `Time`  
`spend`.  
,  
,  
,  
(  
).  
.

`chart.js` <https://www.chartjs.org/docs/latest/>.

4. .

## Пример диаграммы





## Выполненные пункты ТЗ

- : « » « ».
- «Stats»  
 , .  
 : «Money» (  
 ), «Transport» (  
 ) «Time-spend» (  
 ).
- «Stats».
- «Money»  
 . : «

- «Money»
- ?».
- ,
- «Transport»
- »,
- ».
- : «Time-Spend»
- »,
- »,
- ».

## Киноман

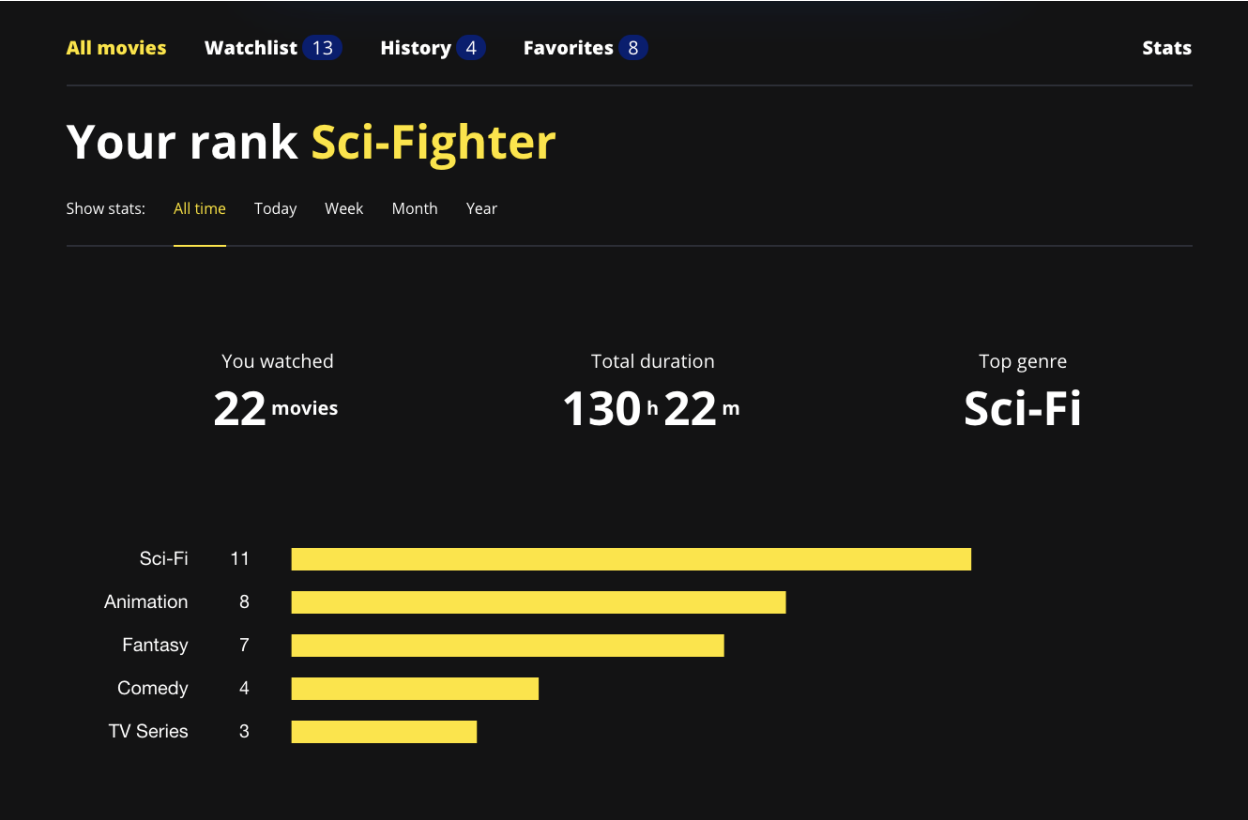
### Смена экранов

1. markup.
  2. main.js
  3. AbstractComponent show/hide
  4. PageController
  5. main.js
- уже
- visually-hidden,
- visually-hidden
- visually-hidden
- 

### Безжалостная статистика

1. `chart.js` `npm` `chart.js` `chartjs-plugin-`  
`datalabels`.
2. `chart.js` `chartjs-plugin-datalabels`,  
`import`.
3. `chart.js`,  
.
- `chart.js` <https://www.chartjs.org/docs/latest/>.
4. .
5. : «You watched»  
(  
(  
, «Total duration» (  
, «Top genre» (  
).

## Пример диаграммы



# Выполненные пункты ТЗ

- : « » « ».
- , .
- , « » :
- , ;
- ;
- ;
- .
- :
- All time;

- Today;
- Week;
- Month;
- Year.

- ,  
.
- ( , «0»,  
( ) — «-».