DOM I. Tinder

1. Разминаем пальцы

Возьмите файлы с готовой разметкой и стилями по этим ссылкам:

[HTML разметка](https://learn.paradox.red/js/task/task_5/first/index.html)

[CSS стили](https://learn.paradox.red/js/task/task_5/first/style.css)

Там в HTML файле уже содержится готовая разметка. Вам же нужно, используя методы DOM API, написать код для создания точно такой же разметки (всё, что внутри body) поэлементно, через метод *createElement*.

1. Верните всю разметку макета обратно в HTML, кроме разметки карточки (*div.card* и весь внутренний набор разметки).

Напишите скрипт, который будет через каждый 2000 мс создавать и добавлять новую карточку, пока их не станет 6 штук. Используйте встроенную функцию *setInterval*. После создания 6 карточек цикл с таймером нужно деактивировать.

Причём, при каждом новом добавлении картинки *img*, к ссылке в неё нужно добавлять уникальный параметр – это может быть счётчик, или просто рандомное число. Чтобы ссылки на картинки выглядели следующим образом:

<https://thispersondoesnotexist.com/image?1>

<https://thispersondoesnotexist.com/image?2>

<https://thispersondoesnotexist.com/image?3>

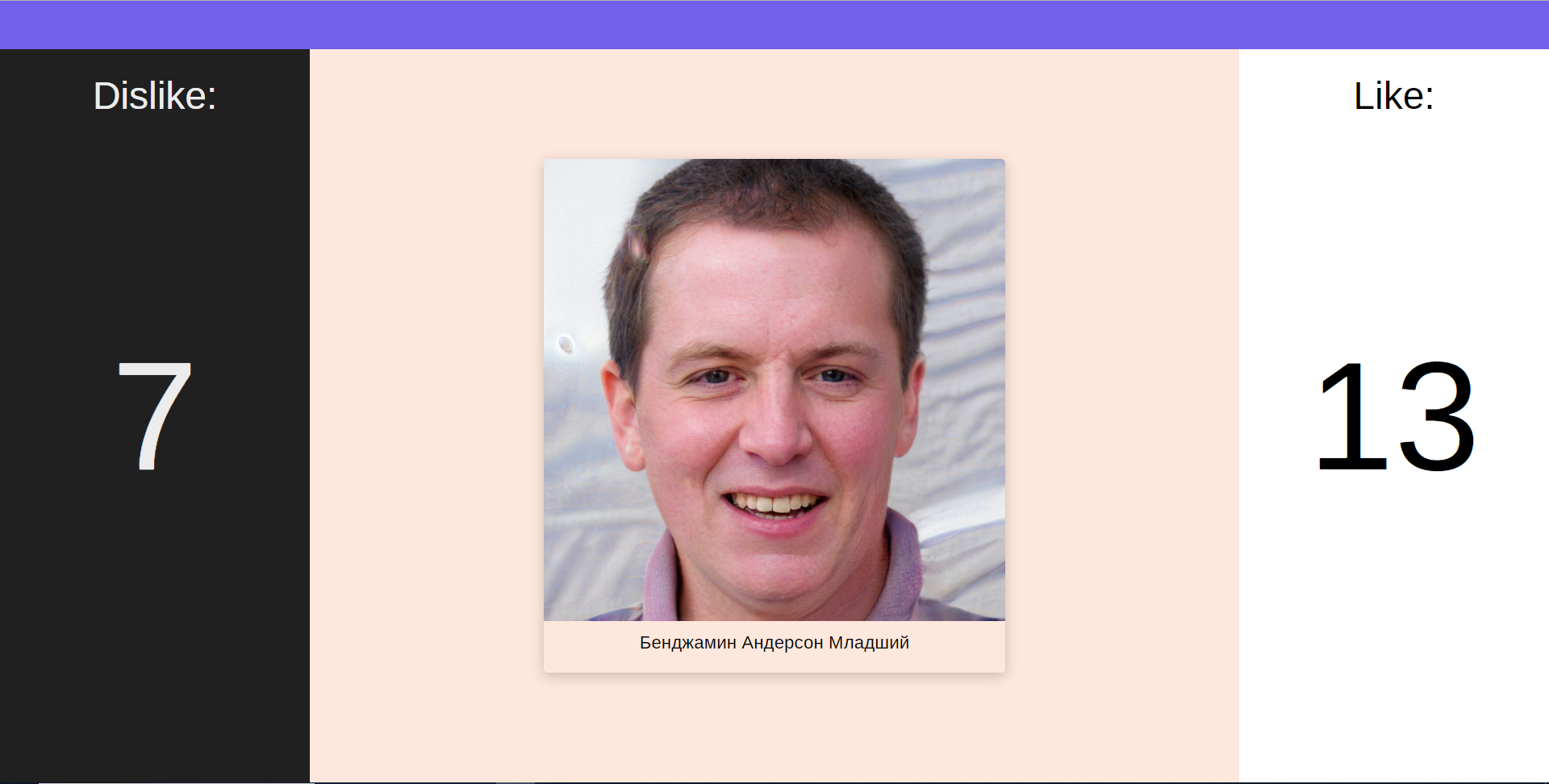
и т.д.

1. Разработайте функцию для генерации случайных имён. Для этого вам нужно иметь массив имён и массив фамилий, из которых функция будет формировать случайное имя.

Используйте эту функцию, чтобы помещать случайное имя на текст для каждой карточки.

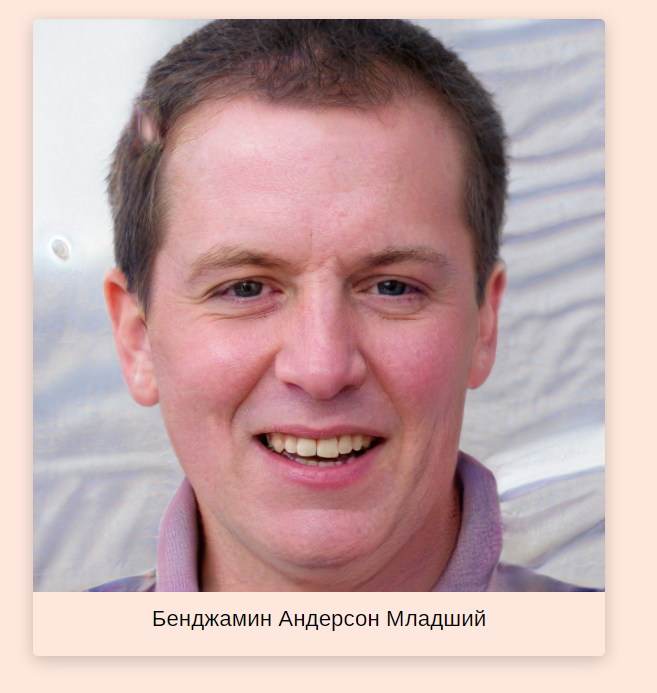
1. Добавьте обработчик события *click* для каждой карточки, по которому эта карточка должна удаляться с экрана, и после этого, через 5000 мс – появляться новая. Для создания отложенного действия используйте встроенную функцию *setTimeout*.
2. Измените разметку таким образом, чтобы на экране оставалась одна большая карточка посередине, а с двух сторон от неё, по бокам, находились две панели.

Пример того как должно получиться на картинке ниже:



В общем, должно получиться что-то вроде Tinder-а, приложения, где можно лайкать/дизлайкать постоянно меняющиеся фотографии.

Идея в том, что при клике на правую половину изображения должен засчитывать лайк, а по нажатию на левую половину изображения – дизлайк. Это нужно сделать с помощью двух абсолютно позиционированных невидимых *div*-а. Один на левой стороне картинки, другой – на правой.



И на эти невидимые дивы нужно повесить обработчики, которые будут менять изображение и увеличивать счётчик лайков/дизлайков.

В соответствии со счётчиком должны обновляться цифры на боковых панелях.

Картинку можно менять просто изменяя ссылку у *img* в атрибуте *src*, добавляя рандомный параметр запроса.

\*Дополнительное задание: сделать так, чтобы состояние счётчика лайков-дизлайков сохранялось в браузере (в cookie или локальном хранилище), и при повторном посещении страницы восстанавливалось оттуда.