

PLAN

Описание продукта:

Лэндинг-пэйдж кинотеатра с возможностью заказа билетов.

Функциональные требования:

Структура лэндинг-пэйдж:

- ✓ Шапка с геолокацией
- ✓ Маркетинговый блок с интерактивными элементами
- ✓ Блок с таблицей — списком фильмов
- ✓ Блок с картой расположения кинотеатра
- ✓ Подвал с контактной информацией

Пользовательские use cases:

- ✓ При посещении лэндинга автоматически определяется регион пользователя и выводится в шапке сайта
- ✓ На маркетинговом блоке размещены три круглых элемента-иконки с подписью (преимущества кинотеатра)
- ✓ При передвижении мышью над преимуществом — иконка реагирует с использованием анимации
- ✓ Одно из преимуществ имеет отношение к интерьеру кинотеатра. При наведении на него мышью всплывает подсказка - «нажми меня, чтобы увидеть интерьер»
- ✓ После нажатия на преимущество всплывает галерея с фотографиями кинотеатра

Выпускной проект по курсу

«Основы web-разработки с применением CSS, HTML, PHP и JavaScript»

- ✓ В блоке со списком фильмов (таблица) предусмотрены колонки: чекбокс (отметить), изображение, наименование, год выпуска, описание
- ✓ Пользователь может сортировать и фильтровать таблицу по наименованию фильма и времени показа
- ✓ Пользователь отмечает в таблице фильм и вызывает форму обратной связи для оформления билетов на него
- ✓ Форма обратной связи представляет из себя всплывающий рор-уп с полями: наименование фильма (подтягивается в соответствии с выбором в таблице), имя, e-mail, загрузка купона на скидку (файла); динамический блок с возможностью добавления строк: ряд (селект), место(селект) билета
- ✓ После заполнения и отправки формы пользователю приходит письмо с благодарностью о совершенном заказе и забронированном месте. Администратору приходит письмо-уведомление
- ✓ На данной стадии реализации проекта не предусмотрена проверка ранее заказанных мест. Позволяем пользователям заказывать все места

Use cases администратора сайта:

- ✓ Для административной области лэндинг пэйдж выделен отдельный URL
- ✓ Поступающие от пользователей заявки сохраняются в базу данных
- ✓ Администратор сайта может просматривать / редактировать / удалять сохраненные заявки
- ✓ На данной стадии реализации проекта не предусмотрен раздел авторизации для административной области лэндинг-пэйдж

Нефункциональные требования:

- ✓ Дизайн лэндинг-пэйдж соответствует дизайну, предоставленному заказчиком (преподавателем) в формате *.psd

**Выпускной проект по курсу
«Основы web-разработки с применением CSS, HTML, PHP и JavaScript»**

- ✓ Дизайн может быть изменён по взаимному соглашению с заказчиком (консультантом курса)
- ✓ Вёрстка лэндинг-пэйдж корректно отображается в современных desktop браузерах: Google Chrome, Mozilla Firefox, Safari, Edge (последние версии)
- ✓ Вёрстка лэндинг-пэйдж корректно отображается в современных мобильных браузерах: iOS Safari, Chrome for Android (последние версии)
- ✓ Вёрстка адаптивная с применением методологии БЭМ, на flex элементах и CSS grid. Возможно использование фреймворка Bootstrap4
- ✓ Вёрстка лэндинга осуществляется итеративно:
 - сначала фиксированная верстка
 - переделывается на резиновую
 - переделывается на адаптивную с использованием media-query
- ✓ Так же используется итеративный подход с точки зрения применения разных подходов к верстке:
 - 1-я итерация: шапка — float, маркетинговый блок — блочная, подвал — CSS grid
 - 2-я итерация: float и блочная верстка переделываются на flex
- ✓ Особое внимание уделить адекватному отображению таблицы с фильмами при отображении на мобильных устройствах:
 - скрывание колонок «описание» и «год выпуска» на экранах менее 720px
 - горизонтальная прокрутка таблицы на экранах менее 480px
- ✓ Там, где это возможно, валидация полей формы обратной связи осуществляется стандартными средствами HTML5
- ✓ Анимация, где возможно, должна быть реализована средствами CSS3
- ✓ Интерактивное взаимодействие с пользователем (всплытие окна, другая анимация, нереализуемая средствами CSS3) - с использованием JavaScript и библиотеки jQuery
- ✓ Информационный блок расположения кинотеатра (карта) выполнен с использованием сервиса Яндекс.Карты

**Выпускной проект по курсу
«Основы web-разработки с применением CSS, HTML, PHP и JavaScript»**

- ✓ Для определения геолокации используется внешний ресурс сервиса Яндекс.Карт, работающий в соответствии с REST:
<https://tech.yandex.ru/maps/jsapi/doc/2.1/ref/reference/geolocation-docpage/>
- ✓ Для вывода списков фильмов используется внешний ресурс, работающий в соответствии с REST: <https://www.themoviedb.org/documentation/api/discover> (endpoint: /discover/movie?sort_by=popularity.desc)
- ✓ Серверная часть реализована с использованием языка программирования PHP 7.2, без использования фреймворков.
- ✓ Отправка писем реализована с помощью php mail
- ✓ База данных SQLite3
- ✓ Для доступа к базе данных используется расширение для PHP - PDO
- ✓ Запросы к базе данных на вставку или редактирование данных должны быть экранированы с использованием prepare statements
- ✓ Сборка и подключение стилей, скриптов js к проекту осуществляется с применением webpack. Возможно использование сторонних библиотек из репозитория npm
- ✓ Обязательно использование языковых средств JavaScript из стандарта ES6.
- ✓ Кодовая база JavaScript реализуется в два этапа:
 - без использования возможностей ES6
 - переписывается с поддержкой ES6 + подключение babel.js
- ✓ Обязательно следование рекомендациям преподавателя и консультантов по стилю оформления кода (именование функций, переменных, классов, методов) и структуре кода (ООП, отсутствие дублирования, логическая пофайловая разбивка проекта и именование файлов)

Цели и задачи обучения:

- ✓ Познакомиться и изучить основы языков программирования и технологий: HTML5, CSS3, JavaScript, PHP+PDO, SQL, SQLite3, REST, Bootstrap4, babel.js, jQuery, webpack, npm, Adobe Photoshop, YandexMap Api, Git

Выпускной проект по курсу
«Основы web-разработки с применением CSS, HTML, PHP и JavaScript»

- ✓ Познакомиться и изучить инструментарии разработчика:
IDE VsCode с плагинами для работы в Git, консоль браузера GoogleChrome
- ✓ Понимать отличия и уметь реализовывать: фиксированную, резиновую, адаптивную верстку сайта
- ✓ Понимать отличия и уметь верстать, используя подходы: float, блочная, flex
- ✓ Познакомиться со спецификацией REST и получением информации со сторонних ресурсов
- ✓ Познакомиться с серверным программированием и основами программирования на PHP
- ✓ Научиться взаимодействовать с серверной частью из клиентского (браузерного) приложения
- ✓ Познакомиться с основами реляционных баз данных и языка SQL
- ✓ Научиться выводить динамическую информацию из базы данных
- ✓ Размещать сайт на веб-хостинге
- ✓ Привить культуру программирования: написания поддерживаемого, расширяемого и понятного кода.
- ✓ Привить умение работы в сжатые сроки, с большим количеством новой информации, документацией. Умению приоретизировать задачи и верно определять ситуации, где уместен перфекционизм, где же необходимо сделать быстро. Ознакомление с методологией MoSCoW