

**Экзаменационное задание по дисциплине
«Основы веб-технологий»
для второго курса по направлению
«Информатика и вычислительная техника»
(профиль «Системная и программная инженерия»)**

1. Основные характеристики экзаменационного задания

Экзаменуемый должен:

Знать:

- Стандарт HTML5;
- Стандарты CSS3;
- Протокол HTTP;
- Bootstrap;
- JavaScript (базовый уровень);
- Git — распределённую систему управления версиями (базовый уровень);
- Основные принципы устройства и функционирования сети Интернет.

Уметь:

- Разрабатывать веб-сайты;
- Программировать на JavaScript;
- Разбираться в документации по используемым технологиям (как в англоязычной, так и в русскоязычной).

Владеть:

- Навыками разработки веб-сайтов;
- Навыками работы с системами управления версиями;
- Поиском информации и рецептов в интернете.

Для оценки степени освоения студентом приведённых знаний, умений и навыков используется 50-балльная шкала. **Минимальные требования и критерии оценивания различных частей экзаменационного задания приведены в приложении № 3.**

2. Общие требования по организации работы и оформлению

- Использовать систему управления версиями Git. Создать отдельный репозиторий для проекта на одной из платформ онлайн-хостинга репозитория (GitHub, GitLab, Bitbucket и т. д.). Код проекта должен быть размещён в ветке main.
- Во время разработки сайта не забывать делать «фиксации» («коммиты», commit) после выполнения каждой минимальной части работы. Сопровождать «фиксации» осмысленными комментариями.
- Необходимо выложить ваш проект на [хостинг университета](#). При использовании хостинга университета нужно использовать домен webdev-exam-2023-1-*****, где вместо звёздочек подставить случайный набор латинских букв и цифр. Такой домен не

смогут подобрать ваши одногруппники, следовательно не смогут скопировать наработки. **Заранее проверьте возможность подключения к хостингу. При возникновении проблем обратитесь к преподавателю.**

- Для отправки проекта на проверку в [задании](#) в СДО Московского Политеха нужно указать ссылку на ваш репозиторий (**не забудьте предоставить доступ преподавателю в случае, если репозиторий закрытый**) и ссылку на ваш проект, выложенный на хостинг университета (если вы выложили на другой хостинг, необходимо указать причину).
- **Имена пользователей преподавателей для предоставления доступа к репозиторию:**
 - Крузалов Алексей Сергеевич: akruzhalov (GitHub, GitLab)
 - Белич Владимир Сергеевич: Belichvladimir (GitHub);
 - Энгиноева Диана Хизировна: enginoevadiana (GitHub), dianaenginoeva (GitLab)
- При оформлении кода необходимо учитывать следующее.
 - Код должен быть написан лесенкой.
 - Переменные, классы, стили и т. п. должны быть названы осмысленно.
 - Все стили и JavaScript должны быть вынесены в отдельные файлы.
 - При необходимости пояснить какой-либо спорный момент или решение — использовать комментарии.
 - При оформлении HTML-кода необходимо использовать расширение [HTMLHint](#). Файл с конфигурацией выложен в разделе «Общая информация о курсе» на главной странице курса в СДО Московского Политеха.
 - При оформлении кода на JavaScript необходимо использовать расширение [ESLint](#). Файлы с конфигурацией и инструкцией по установке выложены в разделе «Общая информация о курсе» на главной странице курса в СДО Московского Политеха.

3. Задание на экзамен

- Разработать сайт для поиска пешеходного туристического маршрута по Москве¹ и оформления заявок на предоставление услуг гида.
- Разработать личный кабинет, позволяющий пользователю управлять своими заявками на предоставление услуг гида.

3.1 Вёрстка

Главная страница

На главной странице должны располагаться следующие компоненты:

- Шапка сайта с заголовком, логотипом и блоком навигации с ссылкой для перехода в личный кабинет (navbar);
- Область для всплывающих уведомлений (alerts);
- Блок с изображением и описанием компании;
- Блок с информацией о достоинствах вашей компании (структура этого блока выбирается по усмотрению студента, однако обязательным условием является использование Bootstrap Grid System);

¹ При разработке задания был использован датасет «[Перечень пешеходных туристических маршрутов Москвы](#)» с портала открытых данных Правительства Москвы.

- Блок с данными о пешеходных маршрутах города Москвы, предоставляющий возможность выбрать один из маршрутов для последующего заказа услуг гида;
- Форма для поиска маршрутов по названию (текстовое поле) и по достопримечательности, через которую должен проходить маршрут (селект);
- Навигационная панель, позволяющая просматривать список маршрутов постранично (пагинация);
- Блок со списком доступных гидов, работающих на выбранном маршруте, с возможностью выбора (отображается после выбора маршрута, в заголовок раздела динамически подставляется название выбранного маршрута);
- Блок с кнопкой «Оформить заявку», по нажатию на которую открывается модальное окно с формой для ввода деталей заказа и расчётом итоговой стоимости;
- Форма для поиска гидов по знанию иностранных языков (селект) и опыту работы (от и до);
- Футер с контактными данными.

Дополнительно на главной странице могут присутствовать следующие элементы²:

- Форма с полем ввода адреса начальной точки;
- Интерактивная карта Москвы (2ГИС).

Примерный макет главной страницы приведён ниже. Необязательные компоненты выделены на макете серой пунктирной рамкой.

В модальном окне для оформления заявки должны присутствовать:

- Название выбранного маршрута (без возможности редактирования);
- ФИО выбранного гида (без возможности редактирования);
- Поле для выбора даты экскурсии (input:date);
- Поле для выбора времени начала экскурсии (input:time);
- Поле для выбора длительности экскурсии (select);
- Поле для указания количества человек в группе (input:number);
- Блок с возможностью указать дополнительные опции заказа (input:checkbox);
- Блок с автоматическим расчётом и отображением итоговой стоимости на основе введённых данных;
- Кнопки «Отправить» и «Отменить».

Примерный макет модального окна для оформления заявки приведён ниже.

Личный кабинет

На странице личного кабинета должны располагаться следующие элементы:

- Шапка сайта с заголовком, логотипом и блоком навигации с ссылкой на главную страницу (navbar);
- Область для всплывающих уведомлений (alerts);
- Таблица с данными заявок на предоставление услуг гида. В таблице должны быть:
 - Порядковый номер заявки в отображаемом списке, название маршрута, дата экскурсии, итоговая стоимость;
 - Блок с кнопками (иконками) «Подробнее», «Редактировать» и «Удалить». При нажатии на кнопку «Подробнее» должно открываться модальное окно с расширенной информацией выбранной заявки. В модальном окне должна


² См. раздел 5.

присутствовать информация об используемых опциях: название, описание и размер скидки/надбавки (вычисляется динамически для каждой заявки). При нажатии на кнопку «Редактировать», должна появляться форма редактирования (модальное окно). При нажатии на кнопку «Удалить», заявка должна удаляться (через модальное окно подтверждения удаления).

- Навигационная панель, позволяющая просматривать список заявок постранично (пагинация);
- Футер с контактными данными.

Примерные макеты страницы личного кабинета и модальных окон просмотра, редактирования и удаления заявки приведены ниже.

Макет страницы личного кабинета












Название компании

Экскурсии по Москве

[Вернуться на главную](#)

Текст сообщения

Оставленные заявки

№	Название маршрута	Дата	Стоимость	
1	Забавушка, музей народной игрушки	10.12.2022	5600	  
2	Смотровая площадка Воробьевы горы	9.12.2022	3200	  
3	Парк "Зарядье"	17.08.2023	2500	  

«

1

2

3

4

5

6

7

8

9

»

← Older

Newer →

[Contacts](#)

Макет модального окна просмотра заявки

×

Заявка номер 3737

ФИО гида: Афанасьева Полина

Название маршрута: Воробьевы горы

Дата экскурсии: 12.01.2022

Время начала: 09:00

Длительность экскурсии: 2 часа

Количество человек: 5

Дополнительные опции:

Название опции

Описание опции

Размер скидки/надбавки

Итоговая стоимость: 4580р

Макет модального окна редактирования заявки

×

Редактирование заявки

ФИО гида: Оксана Лебедева

Название маршрута: Парк "Зарядье"

Дата экскурсии:

17.08.2023

📅

Время начала:

17:00

Длительность экскурсии:

2 часа

▼

Количество человек:

5

Дополнительные опции:

☒ Название опции

☐ Название опции

Описание опции

Описание опции

Размер скидки/надбавки

Размер скидки/надбавки

Итоговая стоимость: 2500 рублей

ОТМЕНИТЬ

СОХРАНИТЬ

Макет модального окна удаления заявки

Вы уверены, что хотите удалить заявку?

Нет

Да


Макет модального окна оформления заявки

Оформление заявки


×

ФИО гида: Афанасьева Полина

Название маршрута: Воробьевы горы

Дата экскурсии: 12.01.2022 

Время начала: 17:00

Длительность экскурсии: 1 час 

Количество человек: 5

Дополнительные опции:

<input checked="" type="checkbox"/> Название опции	<input type="checkbox"/> Название опции
Описание опции	Описание опции
Размер скидки/надбавки	Размер скидки/надбавки

Итоговая стоимость: 4580p

ОТМЕНИТЬ

ОТПРАВИТЬ

Макет главной страницы

Название компании

Экскурсии по Москве

Личный кабинет

Текст сообщения

Слоган/информация о компании

Заголовок

Текст

Текст

Текст

Текст

Текст

Текст

Текст

Текст

Откуда вы хотите начать путешествие:

Введите адрес

Начать

Список пешеходных маршрутов

Поиск по названию

Основной объект

Название	Описание	Основные объекты
Название	Описание	объект, объект <div>Выбрать</div>
Название	Описание	объект, объект <div>Выбрать</div>
Название	Описание	объект, объект <div>Выбрать</div>
Название	Описание	объект, объект <div>Выбрать</div>

123456789

Order

Next

Доступные гиды по маршруту **Название маршрута**

Язык экскурсии

Опыт работы

от

до

	ФИО	Языки	Опыт работы	Стоимость услуг в час	
	ФИО	русский	3 года	Стоимость/час	Выбрать
	ФИО	английский	5 лет	Стоимость/час	Выбрать
	ФИО	русский, английский	4 года	Стоимость/час	Выбрать
	ФИО	французский, английский	8 лет	Стоимость/час	Выбрать

© 2022

Требования к оформлению и содержанию страниц

- Содержимое главной страницы заполняется по усмотрению студента с учётом индивидуального задания, представленного в приложении 1, соответствующего варианту задания. Содержимое блока с информацией о компании, а также футера заполняется по усмотрению студента (изображение и описание должны соответствовать тематике сайта). Навигационная панель должна содержать название и логотип, подобранные в соответствии с тематикой сайта. Таблицы с данными о маршрутах, гидах и заявках должны заполняться при помощи AJAX-запросов к специальному API (см. раздел 4).
- Дополнительные опции для оформления заявки заполняются с учетом индивидуального задания (см. приложение 1)
- В оформлении должны быть использованы цвета из цветовой схемы, соответствующей варианту вашего задания из приложения 2.
- Страница должна быть адаптирована под просмотр на мобильных устройствах. Формат отображения на мобильном устройстве по усмотрению студента, но все компоненты должны отображаться корректно, текст должен быть читабельным, и не должно быть горизонтальной полосы прокрутки.

Требования к оформлению кода и используемым технологиям

- Для вёрстки сайта необходимо использовать фреймворк [Bootstrap](#) версии 5.3. Каркас сайта (сетка), оформление элементов интерфейса и виджеты реализуются с помощью Bootstrap и его компонентов.
- Не приветствуется создание собственных стилей в случае, если требуемого результата можно добиться с использованием стандартных инструментов Bootstrap, описанных в разделе Утилиты (Utilities).
- **Код сайта должен быть написан самостоятельно.** Не допускается использования готовых образцов и решений из интернета. Можно использовать примеры — но переписать их по своему, используя Bootstrap и свои стили. При обнаружении в работе некорректных заимствований кода, оценка за работу будет снижена пропорционально объёму выявленных заимствований.

3.2 Реализовать скрипты для взаимодействия с API

Для реализации функциональности оформления заявки и личного кабинета вам предоставляется CRUD API (см. раздел 4). Он реализует четыре базовые операции по работе с данными: создание (англ. Create), чтение (англ. Read), изменение (англ. Update) и удаление (англ. Delete). Вам необходимо написать скрипт на языке программирования JavaScript, реализующий взаимодействие вашей страницы с этим API посредством AJAX-запросов (можно использовать XMLHttpRequest или fetch) и позволяющий пользователю вашего сайта оформлять заявки на оказание услуг гида, а также просматривать существующие заявки, редактировать и удалять их.

Создание заявки реализуется на главной странице во время выбора маршрута, гида и оформления заявки. Просмотр, изменение и удаление должны быть реализованы в личном кабинете.

3.2.1 Загрузка и отображение списка записей

- Загрузка данных должна осуществляться посредством GET-запросов на URL, приведённые в разделах 4.2 (маршруты), 4.3 (гиды), 4.4 (заявки). По умолчанию в

качестве ответа на этот запрос возвращаются все доступные данные в формате JSON (возвращаемый набор полей для каждой записи можно найти в разделе 4).

- Полученные данные должны быть размещены на странице (см. раздел 3.1). Набор отображаемых полей зависит от типа записи (маршрут, гид, заявка) и определяется приведёнными макетами. В случае, если значение какого-либо поля слишком длинное (например, описание маршрута) допускается сокращённое отображение содержимого этого поля и использование всплывающих подсказок ([tooltip](#)) для просмотра полного значения.
- Так как API не предоставляет возможности осуществить пагинацию данных на стороне сервера, вам необходимо реализовать пагинацию на стороне клиента с помощью JavaScript (для маршрутов и заявок). Максимальное количество записей, расположенных на одной странице равно 10 – для маршрутов и 5 – для заявок. Для отображения кнопок переключения между страницами необходимо использовать [стандартный компонент Bootstrap](#).

3.2.2 Редактирование записи

- При нажатии на кнопку редактирования заявки должно появляться модальное окно с формой для редактирования. Набор полей формы и типы элементов представлены на макете в разделе 3.1.
- Заголовок модального окна «Редактирование заявки».
- Значения полей формы должны быть автоматически установлены в соответствии со значениями полей редактируемой записи.
- При нажатии на кнопку «Сохранить» должен быть произведён PUT-запрос на URL, приведённый в разделе 4.6.
- Если операция была успешно выполнена, необходимо разместить в области уведомлений сообщение об удачном завершении операции, очистить поля формы, а также обновить данные соответствующей записи в таблице. Если при выполнении операции произошла ошибка, необходимо вывести уведомление об ошибке.
- Для оформления уведомлений необходимо использовать стандартное цветовое оформление фреймворка Bootstrap, соответствующее типу уведомления. Уведомления должны автоматически исчезать по прошествии 5 секунд после их появления.

3.2.3 Удаление записи

- При нажатии на кнопку удаления записи должно появляться диалоговое окно подтверждения операции удаления. Окно должно содержать вопрос «Вы уверены, что хотите удалить заявку?» и кнопки «Да» и «Нет».
- При нажатии на кнопку «Нет» происходит закрытие диалогового окна. При нажатии на кнопку «Да» происходит отправка DELETE-запроса на URL, приведённый в разделе 4.7. В качестве ответа сервер вернёт результат выполнения операции (см. формат ответа в разделе 4).
- Если операция была успешно выполнена, необходимо удалить данные соответствующей записи из таблицы.
- Статус выполнения операции должен быть отображён пользователю с помощью уведомления (аналогично операции редактирования).

3.2.4 Создание записи (оформление заявки)

- При нажатии на кнопку «Оформить заявку» должно появляться модальное окно с формой для создания заявки.

- Заголовок модального окна «Оформление заявки».
- Набор полей формы аналогичен форме для редактирования, описанной выше. Во избежание дублирования кода рекомендуется использовать одно модальное окно для операций создания и редактирования записей.
- При нажатии на кнопку «Отправить» должен быть произведён POST-запрос на URL, приведённый в разделе 4.5. В качестве ответа сервер вернёт результат выполнения операции (см. формат ответа в разделе 4.1).
- Статус выполнения операции должен быть отображён пользователю с помощью уведомления (см. выше).

3.3 Система оформления заказа

3.3.1 Поиск маршрутов

На главной странице выводится таблица туристических маршрутов Москвы. Таблица должна содержать следующие столбцы:

- Название маршрута;
- Краткое описание;
- Основные объекты;
- Столбец с кнопкой выбора маршрута.

Над таблицей на основании всех полученных с сервера данных должен автоматически формироваться фильтр с выбором основного объекта маршрута. Реализуется в виде выпадающего списка (select). Должен быть предусмотрен пункт «Не выбрано». Также в форме для поиска должно быть предусмотрено текстовое поле для поиска по названию. При смене значений формы нужно обновлять таблицу с учетом введенных данных. Пустые значения полей игнорируются, то есть по ним не проводится поиск.

При нажатии на кнопку выбора маршрута, строка, соответствующая выбранному маршруту в таблице, должна быть выделена другим цветом. Выбор не должен сбрасываться при переходе на другую страницу или при выполнении поиска.

3.3.2 Выбор гида

При выборе маршрута под списком маршрутов подгружается (см. раздел 4.3) таблица доступных для данного маршрута гидов. Таблица должна быть скрыта до момента выбора маршрута. Таблица должна содержать следующие столбцы:

- Картинка профиля (выбирается студентом, может быть одинаковой для всех гидов);
- ФИО;
- Языки, которыми владеет гид;
- Опыт работы;
- Стоимость экскурсионного обслуживания (рублей в час);
- Столбец с возможностью выбора гида (например, кнопкой).

Данные загружаются с помощью API (раздел 4).

Над таблицей должны автоматически формироваться фильтры с выбором языка гида и опыта работы. Фильтры формируются на основании всех полученных с сервера данных. Выбор языка реализуется в виде выпадающего списка (должен быть предусмотрен пункт «Не выбрано»), указание опыта – с помощью текстовых полей «от» и «до». При смене значения

фильтра нужно обновлять таблицу с учетом выбранного значения. Пустые значения полей игнорируются, то есть по ним не проводится поиск.

При нажатии на кнопку выбора гида, строка, соответствующая выбранному гиду в таблице, должна быть выделена другим цветом. Выбор не должен сбрасываться при переходе на другую страницу или при выполнении поиска.

3.3.3 Оформление заявки

Под списком гидов расположена кнопка «Оформить заявку». При нажатии на нее появляется модальное окно с формой. Форма должна содержать:

- Название выбранного маршрута (без возможности редактирования);
- ФИО выбранного гида (без возможности редактирования);
- Поле для выбора даты экскурсии (input:date);
- Поле для выбора времени начала экскурсии (input:time). Доступное время – с 9 до 23 с шагом 0,5 ч.;
- Поле для выбора длительности экскурсии (select, от 1 до 3 часов);
- Поле для указания количества человек в группе (input:number, целое число от 1 до 20);
- Дополнительные опции заказа в виде чекбоксов (список опций в приложении 1).
- Итоговая стоимость, которая автоматически рассчитывается по формуле (см. раздел 3.3.4);
- Кнопки «Отправить» и «Отменить».

3.3.4 Стоимость заявки

Для выполнения задания необходимо динамически рассчитывать стоимость экскурсионного обслуживания по следующей формуле.

$$\text{Price} = \text{guideServiceCost} \times \text{hoursNumber} \times \text{isThisDayOff} + \text{isItMorning} + \text{isItEvening} + \text{numberOfVisitors},$$

где:

- guideServiceCost – стоимость услуг гида за один час;
- hoursNumber – длительность экскурсии в часах;
- isThisDayOff – множитель, отвечающий за повышение стоимости в праздничные и выходные дни. Для будней³ равен 1, для праздничных и выходных дней (сб, вс) – 1,5;
- isItMorning – надбавка за раннее время экскурсии. Для экскурсий, которые начинаются с 9 до 12 часов, равна 400 рублей, для остальных – 0;
- isItEvening – надбавка за вечернее время экскурсии. Для экскурсий, которые начинаются с 20 до 23 часов, равна 1000 рублей, для остальных – 0;
- numberOfVisitors - надбавка за количество посетителей экскурсии:
 - от 1 до 5 человек – 0 рублей,
 - от 5 до 10 – 1000 рублей,
 - от 10 до 20 – 1500 рублей.

³ Чтобы узнать день недели интересующей даты, можно воспользоваться методом `getDay` [объекта даты](#). Список праздников можно найти в актуальном [производственном календаре](#).

4. Описание API

4.1 Модель данных

Маршруты

Название параметра	Тип	Ограничение	Обязательное	Только для чтения	Примечание
id	int			Y	
name	string	max 255 символов		Y	
description	string			Y	
mainObject	string			Y	Достопримечательности маршрута (разделены символом «-»).
coords	JSON			Y	Координаты начала маршрута (массив в формате JSON)
created_at	DateTime			Y	

Пример в формате JSON

```
{
  "coords": [
    37.525949478149,
    55.64294205603
  ],
  "created_at": "2022-12-18T15:12:00",
  "description": "Район, которого нет, район, который хотели занести в наследие ЮНЕСКО и почему, район академический, экспериментальный, художественно-поэтический.\nМаршрут разработан Боровских Майей Владимировной.",
  "id": 20,
  "mainObject": "м. Беляево. Стихограмма Пригова, стритарт Фариды. Руэда - площадь Мартина Лютера Кинга. Верстовой столб. Храм Троицы Живоначальной. Дом Цоя. Яблоневый сад. Ул. Волгина: дом Пригова, Московский Университете МВД. ИБХ Шемякина и Овчинникова, памятник антибиотик. Российский Геологоразведочный университет, скульптуры во дворе, Институт русского языка им Пушкина, сад букв, нулевой километр московского интернета, туалет, галерея Беляево.",
  "name": "Авторский маршрут Боровских М.В. «Герцогство Беляевское»"
}
```

Гиды

Название параметра	Тип	Ограничение	Обязательное	Только для чтения	Примечание
id	int			Y	
name	string	max 255 символов		Y	
workExperience	int			Y	
language	string	max 255 символов		Y	
pricePerHour	int			Y	
route_id	int			Y	

Пример в формате JSON

```
{
  "id": 117,
  "language": "Nepali",
  "name": "Sabina Dixon",
  "pricePerHour": 1913,
  "route_id": 20,
  "workExperience": 6
}
```

Заявки

Название параметра	Тип	Ограничение	Обязательное	Только для чтения	Примечание
id	int			Y	
guide_id	int		Y		
route_id	int		Y		
date	date		Y		Предавать в формате YYYY-MM-DD
time	time		Y		Предавать в формате HH:MM
duration	int		Y		от 1 до 3
persons	int		Y		от 1 до 20

price	int		Y		
optionFirst	bool		Y		Передавать значения 0 или 1
optionSecond	bool		Y		Передавать значения 0 или 1
student_id	int			Y	Идентификатор студента, которому принадлежит заявка

Пример в формате JSON

```
{
  "date": "2022-12-31",
  "duration": 3,
  "guide_id": 5,
  "id": 2,
  "optionFirst": false,
  "optionSecond": true,
  "persons": 10,
  "price": 5000,
  "route_id": 5,
  "student_id": 10700,
  "time": "14:00:00"
}
```

Авторизация

Для получения доступа к API необходимо пройти процедуру авторизации. Для авторизации нужно указать в качестве параметра запроса `api_key` значение уникального ключа, который выдаётся каждому пользователю. Ключ представляет собой идентификатор UUIDv4, который является случайным 16-байтным номером (например, 123e4567-e89b-12d3-a456-426655440000).

Для получения ключа необходимо перейти по ссылке, которая выложена в разделе с экзаменационным проектом в СДО Московского Политеха.

Обратите внимание, что параметр `api_key` всегда передаётся в строке запроса.

Пользователь может просматривать, редактировать и удалять только свои заявки. В один момент времени в базе данных может быть не более 10 заявок, созданных одним и тем же пользователем.

Если пользователь попытается совершить действие, не пройдя авторизацию, в качестве ответа на его запрос придёт сообщение

```
{'error': 'Для получения доступа к API необходимо пройти процедуру авторизации. Для этого  
нужно передать в запросе персональный API Key.'}
```

При передаче параметров в POST- и PUT-запросах данные должны передаваться в теле запроса в формате `application/x-www-form-urlencoded` или `multipart/form-data` (для этого достаточно при отправке данных формы воспользоваться объектом [FormData](#)).

4.2 Получить список маршрутов

URL	http://exam-2023-1-api.std-900.ist.mospolytech.ru/api/routes
Method	GET
Response	JSON [{item1},{item2},...{itemN}]

4.3 Получить список гидов

URL	http://exam-2023-1-api.std-900.ist.mospolytech.ru/api/routes/{id-маршрута}/guides
Method	GET
Response	JSON [{item1},{item2},...{itemN}]

4.4 Получить список заявок

URL	http://exam-2023-1-api.std-900.ist.mospolytech.ru/api/orders
Method	GET
Response	JSON [{item1},{item2},...{itemN}]

4.5 Добавить заявку

URL	http://exam-2023-1-api.std-900.ist.mospolytech.ru/api/orders
Method	POST
Params	field1=value1, field2=value2, fieldN=valueN Важно передать все обязательные поля
Response	JSON {newItem}

4.6 Редактировать заявку

URL	http://exam-2023-1-api.std-900.ist.mospolytech.ru/api/orders/{id-заявки}
Method	PUT

Params	field1=value1, field2=value2, fieldN=valueN Передавать можно только те поля, которые необходимо обновить
Response	JSON {updateItem}

4.7 Удалить заявку

URL	http://exam-2023-1-api.std-900.ist.mospolytech.ru/api/orders/{id-заявки}
Method	DELETE
Response	JSON {id: ID удаленной записи}

4.8 Посмотреть заявку

URL	http://exam-2023-1-api.std-900.ist.mospolytech.ru/api/orders/{id-заявки}
Method	GET
Response	JSON {Item}

4.9 Получить информацию о гиде

URL	http://exam-2023-1-api.std-900.ist.mospolytech.ru/api/guides/{id-гида}
Method	GET
Response	JSON {Item}

5. Дополнительное задание

Данное задание является опциональным и необязательно к выполнению.

В данном задании вам необходимо добавить на страницу элементы для работы с API 2GIS.

Перед использованием API 2GIS нужно получить ключ доступа. Для этого необходимо заранее заполнить форму на dev.2gis.ru/order.

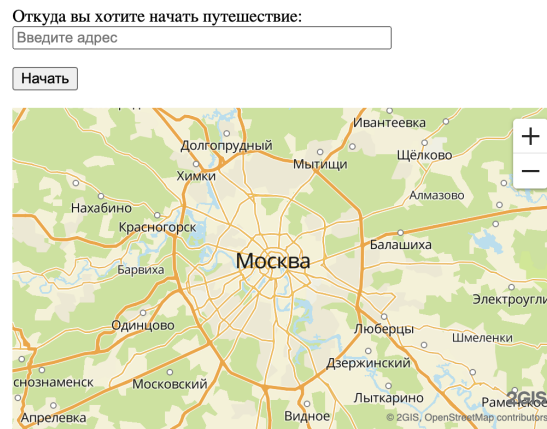
Верстка

Добавить на главную страницу следующие элементы:

1. Форма с полем ввода адреса начальной точки;
2. Интерактивная карта Москвы (2ГИС).

Функционал (API и JavaScript)

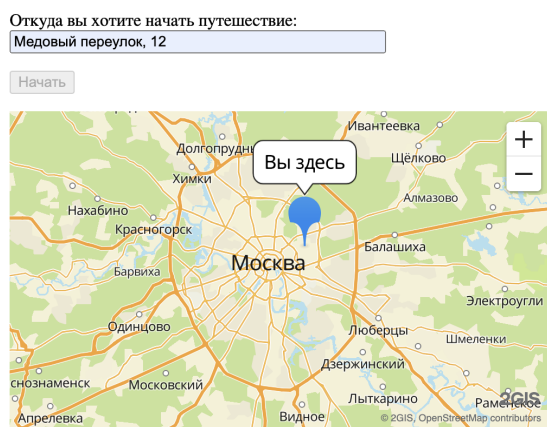
При открытии главной страницы на ней должна быть размещена форма с полем ввода адреса и карта Москвы.



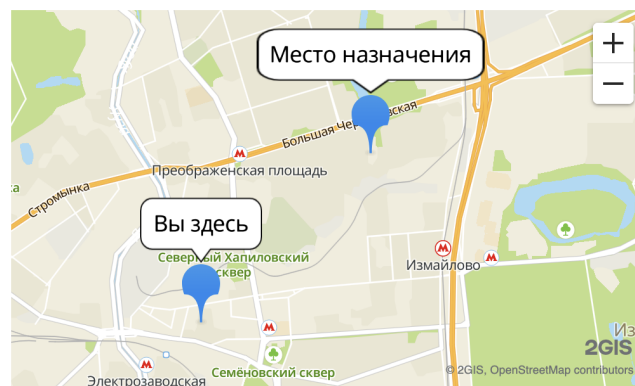
Адрес необходимо вводить в формате: «Улица, номер дома».

Например: «Медовый переулок, 12».

После заполнения и отправки формы, кнопка отправки блокируется, а на карте появляется маркер с введенным адресом.



После выбора маршрута, его начальная точка также появляется на карте в виде маркера.

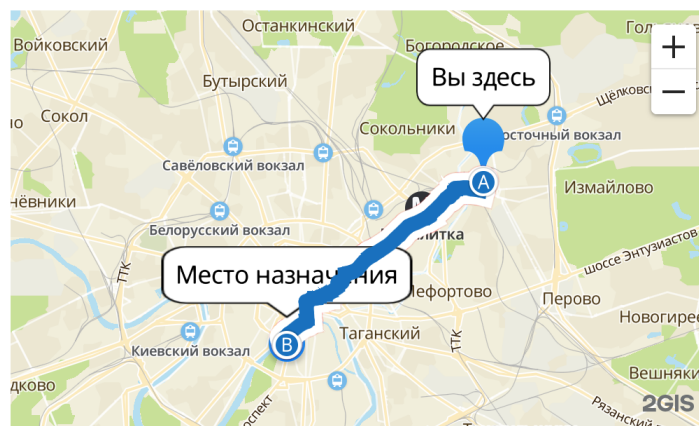


Куда вы хотите отправиться:

Музей **Парк** **Смотровая площадка**

Забавушка, музей народной игрушки

Проложить пешеходный маршрут от одной точки до другой, чтобы он отобразился на карте.



Реализовать функции:

- Добавить на сайт карту с помощью библиотеки MapGL JS API;
- Получить адрес начальной точки с помощью формы;
- Вывести введенный адрес в виде маркера на добавленной карте с подписью «Вы здесь» при помощи преобразования адреса в координаты (Geocoder API);
- При выборе маршрута, выводить его маркер на карте (Geocoder API);
- Проложить на карте пешеходный маршрут от одной точки до другой (Directions API).

API 2GIS

Добавление карты

```
const map = new mapgl.Map('container', {  
  key: 'Your API access key',  
  center: [55.31878, 25.23584],  
  zoom: 13,  
});
```

Документация <https://docs.2gis.com/ru/mapgl/overview>

Добавление маркера с подписью

Пример:

```
const marker = new mapgl.Marker(map, {  
  coordinates: [55.31878, 25.23584],  
  label: {  
    text: "Общага",  
    offset: [0, -75],  
    image: {  
      url: 'https://docs.2gis.com/img/mapgl/tooltip.svg',  
      size: [100, 40],  
      padding: [10, 10, 20, 10],  
    },  
  },  
});
```

Документация <https://docs.2gis.com/ru/mapgl/overview>

Получить координаты по адресу

Пример запроса:

https://catalog.api.2gis.com/3.0/items/geocode?q=Москва, Садовническая, 25&fields=items.point&key=YOUR_KEY

Документация <https://docs.2gis.com/ru/api/search/geocoder/overview>

Проложить пешеходный маршрут между точками

Пример запроса:

```
directions.pedestrianRoute({  
    points: [marker1.getCoordinates(), marker3.getCoordinates()],  
});
```

Документация <https://docs.2gis.com/ru/api/navigation/directions/overview>

Приложение 1. Опции заказа

1. Использовать скидку для школьников и студентов. При выборе чекбокса стоимость уменьшается на 15%.
2. Использовать скидку для пенсионеров. Стоимость уменьшается на 25%.
3. Быстрый выезд гида (в течение часа). Повышает стоимость на 30%.
4. Легкие закуски и горячие напитки во время экскурсии. Увеличивают стоимость на 1000 рублей за каждого посетителя.
5. Тематические сувениры для посетителей. Стоимость увеличивается на 500 рублей для каждого посетителя.
6. Трансфер до ближайших станций метро после экскурсии. Увеличивает стоимость на 25% в выходные дни и на 30% в будние.
7. Интерактивный путеводитель. Увеличивает стоимость в 1,5 раза.
8. Сопровождение сурдопереводчика. Увеличивает стоимость на 15%, если посетителей от 1 до 5, на 25%, если от 5 до 10. При выборе количества посетителей больше 10, данная опция блокируется.

Приложение 2. Цветовые схемы

1. Белый: с синим, красным и черным.
2. Бежевый: с голубым, коричневым, изумрудным, черным, красным, белым.
3. Серый: с цветом фуксии, красным, фиолетовым, розовым, синим.
4. Розовый: с коричневым, белым, цветом зеленой мяты, оливковым, серым, бирюзовым, нежно-голубым.
5. Фуксия (темно-розовый): с серым, желто-коричневым, цветом лайма, зеленой мяты, коричневым.
6. Красный: с желтым, белым, бурым, зеленым, синим и черным.
7. Томатно-красный: голубой, цвет зеленой мяты, песчаный, сливочно-белый, серый.
8. Вишнево-красный: лазурный, серый, светло-оранжевый, песчаный, бледно-желтый, бежевый.
9. Малиново-красный: белый, черный, цвет дамасской розы.
10. Коричневый: ярко-голубой, кремовый, розовый, палевый, зеленый, бежевый.
11. Светло-коричневый: бледно-желтый, кремово-белый, синий, зеленый, пурпурный, красный.
12. Темно-коричневый: лимонно-желтый, голубой, цвет зеленой мяты, пурпурно-розовый, цветом лайма.
13. Рыжевато-коричневый: розовый, темно-коричневый, синий, зеленый, пурпурный.
14. Оранжевый: голубой, синий, лиловый, фиолетовый, белый, черный.
15. Светло-оранжевый: серый, коричневый, оливковый.
16. Темно-оранжевый: бледно-желтый, оливковый, коричневый, вишневый.
17. Желтый: синий, лиловый, светло-голубой, фиолетовый, серый, черный.
18. Лимонно-желтый: вишнево-красный, коричневый, синий, серый.
19. Бледно-желтый: цвет фуксии, серый, коричневый, оттенки красного, желтовато-коричневый, синий, пурпурный.
20. Золотисто-желтый: серый, коричневый, лазурный, красный, черный.
21. Оливковый: апельсиновый, светло-коричневый, коричневый.
22. Зеленый: золотисто-коричневый, оранжевый, салатный, желтый, коричневый, серый, кремовый, черный, сливочно-белый.
23. Салатный цвет: коричневый, желтовато-коричневый, палевый, серый, темно-синий, красный, серый.
24. Бирюзовый: цвет фуксии, вишнево-красный, желтый, коричневый, кремовый, темно-фиолетовый.
25. Электрик красив в сочетании с золотисто-желтым, коричневым, светло-коричневым, серым или серебряным.
26. Голубой: красный, серый, коричневый, оранжевый, розовый, белый, желтый.
27. Темно-синий: светло-лиловый, голубой, желтовато-зеленый, коричневый, серый, бледно-желтый, оранжевый, зеленый, красный, белый.
28. Лиловый: оранжевый, розовый, темно-фиолетовый, оливковый, серый, желтый, белый.
29. Темно-фиолетовый: золотисто-коричневый, бледно-желтый, серый, бирюзовый, цвет зеленой мяты, светло-оранжевый.
30. Черный: с оранжевым, розовым, салатным, белым, красным, сиреневатым или желтым.

По материалам [decor-design](http://decor-design.ru)

Приложение 3. Критерии оценки

- 1) Полностью сверстана главная страница и работает загрузка и отображение списка маршрутов – 15 баллов (необходимо для сдачи проекта).
- 2) Оформление заявки (весь остальной функционал главной страницы) – до 15 баллов.
- 3) Личный кабинет – до 20 баллов.
- 4) Дополнительное задание – до 15 баллов.

Максимальное количество баллов – 50.

Пункт 1 является необходимым для защиты проекта – при его невыполнении проект не засчитывается.