

**Экзаменационное задание по дисциплине**  
**«Основы веб-технологий»**  
**для второго курса по направлению**  
**«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 в интернете.

Для оценки степени освоения студентом приведенных знаний, умений и навыков используется 40-балльная шкала. Минимальные требования и критерии оценивания различных частей экзаменационного задания приведены в приложении № 1.

## 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-2025-1-\*\*\*\*\*, 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:

- Мохначев Виктор Сергеевич: gagashaggy (GitHub), ViktorMokhnachev (Bitbucket)
- Ашрафи Арифа: Arifaa13 (GitHub)
- 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 языковой школы, в которой студенты могут изучать иностранные языки.

#### 3.1 Вёрстка

##### **Компоненты главной страницы:**

- **Шапка сайта:** Заголовок, включающий название сайта (языковой школы), логотип и панель навигации со ссылками на другие разделы (например, «Курсы», «О нас», «Контакты», «Личный кабинет»).
- **Область уведомлений:** место для всплывающих оповещений или уведомлений.
- **Блок описания языковой школы:** раздел, содержащий информацию о службе, ее миссии и образовательных предложениях.
- **Преимущества изучения иностранного языка:** раздел, в котором описываются преимущества изучения иностранного языка; для вёрстки необходимо использовать Bootstrap Grid System.
- **Предлагаемые курсы:** блок, демонстрирующий доступные языковые курсы с возможностью выбора курса для получения дополнительной информации.
- **Форма поиска:** Форма для поиска курсов по названию или уровню (начальный, средний, продвинутый).

- **Пагинация курсов:** навигационная система, позволяющая пользователям просматривать курсы на нескольких страницах.
- **Списки репетиторов:** блок со списком доступных репетиторов языка, динамически отображающий информацию о репетиторах в зависимости от выбранного курса.
- **Кнопка формы заказа:** кнопка, которая открывает модальное окно с формой для запроса сеанса репетиторства или дополнительной информации.
- **Поиск репетиторов:** форма для фильтрации репетиторов по квалификации (выпадающий список) и опыту (количество лет).
- **Футер:** контактная информация и ссылки на социальные сети.

Для получения дополнительных баллов могут быть добавлены следующие компоненты:

- **Карта интерактивных учебных ресурсов:** карта, показывающая места, где учащиеся могут найти ресурсы (например, библиотеки, языковые кафе) для изучения иностранных языков.

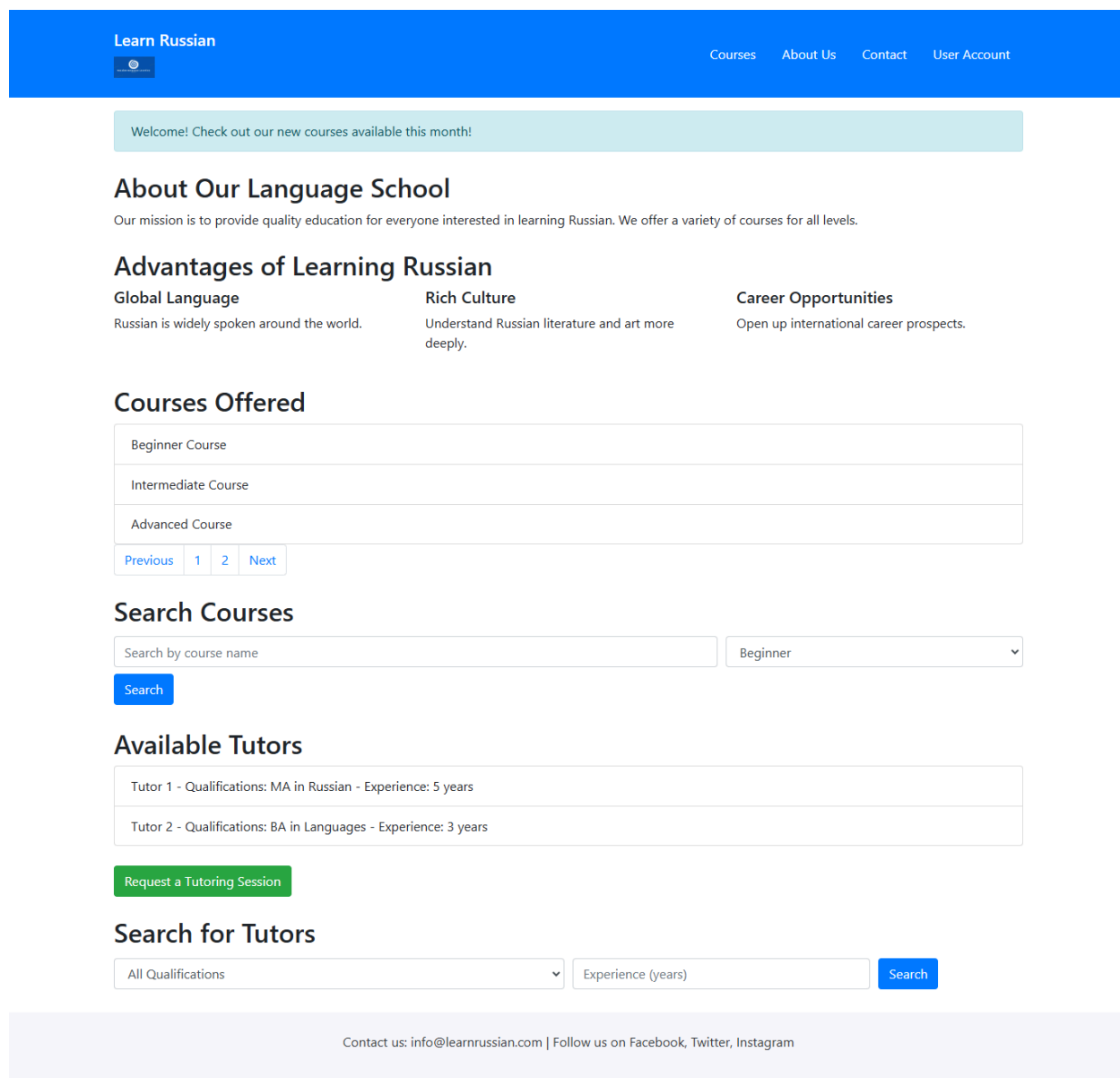


Рисунок 1: Пример макета главной страницы

The image shows a web interface with a dark grey background. In the top left, there is a green button labeled 'Request a Tutoring Session'. On the right, a modal window is open with a blue header containing the text 'Request a Tutoring Session' and a close icon (X). The modal body is white and contains three form fields: 'Your Name' (a single-line text input), 'Email Address' (a single-line text input), and 'Message' (a multi-line text area). At the bottom right of the modal, there are two buttons: a grey 'Close' button and a blue 'Submit' button.

Рисунок 2: Пример запроса на занятие с репетитором

## Личный кабинет

**Страница личного кабинета должна содержать:**

**Шапка сайта:** заголовок, включающий название сайта, логотип и навигационный блок со ссылкой на главную страницу.

**Область уведомлений:** место для всплывающих оповещений или уведомлений.

**Таблица заказов:** Таблица, отображающая:

Порядковый номер заказа, название курса, дата занятия, общая стоимость.

Блоки с кнопками действий: «Подробнее», «Изменить» и «Удалить».

**Кнопка «Подробнее»:** при нажатии открывается модальное окно с расширенной информацией о выбранном заказе (с использованием параметров: название, описание и рассчитанные скидки/надбавки).

**Кнопка «Изменить»:** при нажатии открывается форма редактирования (модальное окно), позволяющая пользователю обновить свой заказ.

**Кнопка «Удалить»:** при нажатии открывается модальное окно подтверждения для удаления заказа.

**Пагинация:** навигационная система, позволяющая пользователям просматривать заказы на нескольких страницах.

**Футер:** контактная информация и ссылки на социальные сети.

### Примеры макетов страницы личного кабинета:

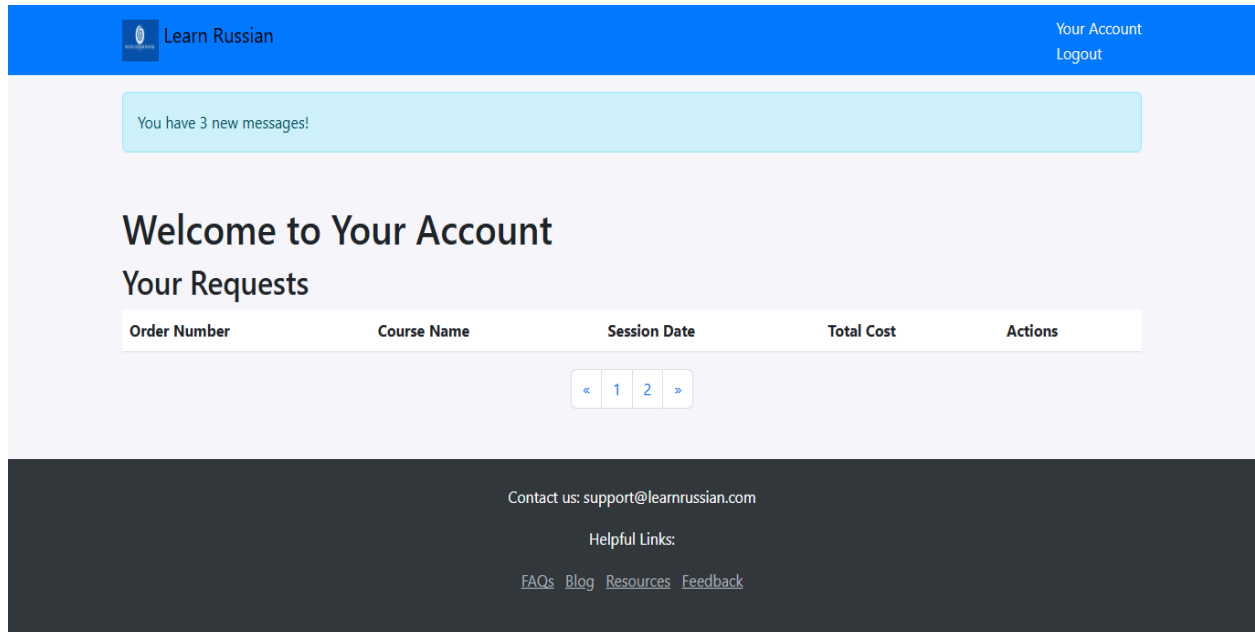


Рисунок 3: Пример макета личного кабинета (без заказа какого-либо курса)

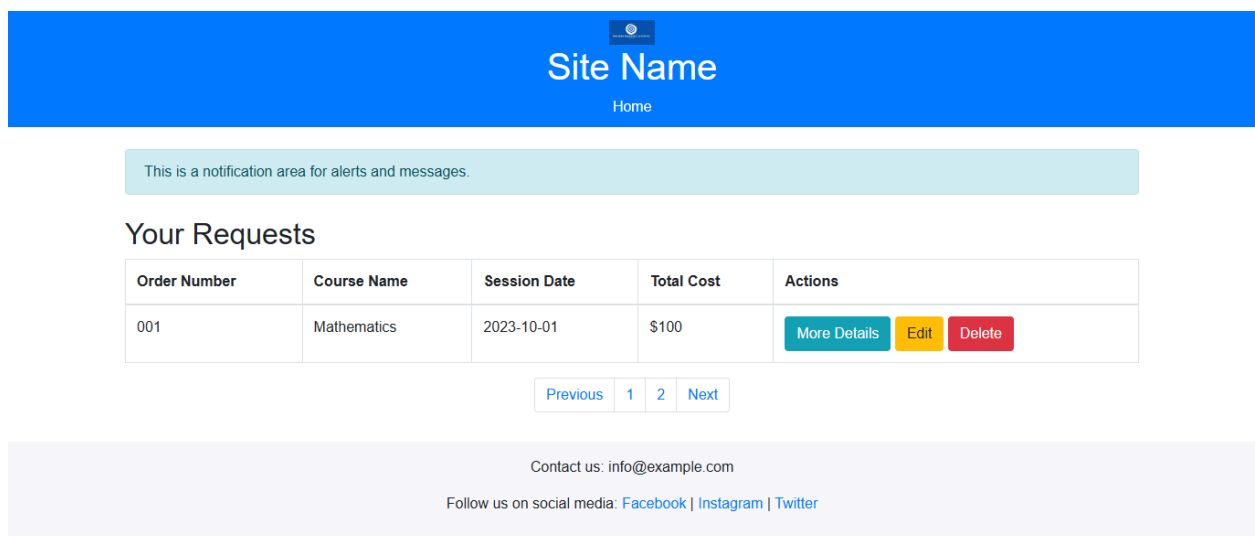


Рисунок 4: Еще один макет личного кабинета (с заказом)

### **Требования к оформлению и содержанию страниц:**

- Содержание главной страницы заполняется по усмотрению студента, при этом оно должно соответствовать теме изучения иностранных языков.
- Футер должен содержать контактную информацию и полезные ссылки, предназначенные для учащихся.
- Панель навигации должна содержать название и логотип сайта, соответствующие образовательной теме.
- Таблицы курсов и репетиторов будут использовать запросы AJAX для связи со специальным API (см. раздел 4).
- Веб-сайт должен быть адаптивным, чтобы все компоненты корректно отображались на мобильных устройствах.
- Цветовое оформление и прочие аспекты дизайна остаются на усмотрение студента.

### **Требования к форматированию кода и используемым технологиям:**

Для вёрстки сайта необходимо использовать фреймворк [Bootstrap](#) версии 5.3. Если в этом нет необходимости, отдавайте предпочтение стандартным утилитам Bootstrap, а не пользовательским стилям.

**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. Не допускается использование готовых образцов, примеров и решений из интернета, AI-инструментов для генерации кода. Можно использовать примеры, но нужно переписать их по-своему – **нельзя использовать чужой код**. При выявлении факта заимствования чужого кода будет снижена оценка за экзамен.

### **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 репетитора, а также просматривать существующие заявки, редактировать и удалять их.

Создание заявки реализуется на главной странице во время выбора курса или репетитора 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 (курсы), 4.3 (репетиторы), 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 (курс, репетитор, заявка) и определяется приведёнными макетами. В случае, если значение какого-либо поля слишком длинное (например, описание школы) 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 курсов и заявок). Максимальное количество записей, расположенных на одной странице равно 5. Для отображения кнопок переключения между страницами необходимо использовать стандартный компонент Bootstrap.

### **3.2.2 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 и типы элементов представлены на макете в разделе 3.3.4.

- При нажатии на кнопку «Отправить» должен быть произведён POST-запрос на URL, приведенный в разделе 4.5. В качестве ответа сервер вернёт результат выполнения операции (см. формат ответа в разделе 4.1).
- Статус выполнения операции должен быть отображен пользователю с помощью уведомления (см. выше).

### **3.2.3 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 3.3.4. Во избежание дублирования кода рекомендуется использовать одно модальное окно для операций создания и редактирования записей.
- 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,

приведенный 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 отображен пользователю с помощью уведомления (аналогично операции редактирования).

### **3.3 Процесс оформления заявки**

#### **3.3.1 Поиск репетитора**

##### **Форма поиска репетитора:**

Отобразить форму поиска со следующими полями:

- Предлагаемые языки
- Уровень языка (начальный, средний, продвинутый)

##### **Таблица результатов поиска:**

Отобразить таблицу со следующими столбцами:

- Имя репетитора
- Уровень языка
- Языки, на которых говорит репетитор
- Опыт (лет)
- Почасовая ставка (руб./час)
- Фото профиля (в API фото отсутствуют, можно поставить изображение-заглушку)
- Кнопка выбора репетитора.

##### **Функциональность поиска:**

- Таблица должна обновляться в режиме реального времени по мере изменения пользователем критериев поиска.
- Пустые значения полей не должны влиять на возможности фильтрации.

#### **3.3.2 Выбор репетитора**

Когда репетитор выбран, строка, соответствующая ему, должна выделяться другим цветом.

[Tutor Search](#) [Home](#) [About Us](#) [Services](#) [Contact](#)

Find a Russian Tutor

Language Level:  

All

Available Days:  

e.g., Monday, Wednesday

Search

Tutor Name	Language Level	Languages Spoken	Experience (years)	Hourly Rate (rubles)	Select Tutor
Anna Ivanova	Beginner	Russian, English	5	1000	Select
Dmitry Petrov	Intermediate	Russian, German	8	1500	Select
Elena Sokolova	Advanced	Russian, French	10	2000	Select

© 2023 Russian Tutor Search | [Privacy Policy](#) | [Contact Us](#)

Рисунок 5: Пример макета для поиска репетитора

### 3.3.3 Подача заявки на языковые курсы

Под списком доступных языковых курсов находится кнопка «Подать заявку». Также кнопка “Подать заявку” может находиться в таблице со списком курсов в отдельном столбце, как у репетиторов на рисунке выше. При нажатии на эту кнопку появляется модальное окно с формой. Форма должна содержать следующие поля:

1. Название выбранного курса (нередатируемое поле).
2. Полное имя преподавателя (нередатируемое поле).
3. Поле для выбора даты начала курса:
  - Поле для выбора даты начала курса (input:date, либо выпадающий список). Доступные значения берутся из API.
4. Поле для выбора времени занятий:
  - Поле для выбора времени занятия (выпадающий список). Недоступно до выбора даты. В каждом значении присутствует время начала занятия, взятое из API, и вычисленное время окончания занятия на основе данных о

продолжительности курса в неделю. При этом при отправке заявки на сервер передаётся только время начала.

5. Продолжительность курса (нередатируемое поле):

- Отображается продолжительность курса в неделях, здесь же отображается вычисленная дата окончания курса (последнего занятия).

6. Поле для указания количества студентов в группе:

- Поле ввода для указания количества студентов в группе (input:number, диапазон от 1 до 20, как целочисленное значение)

7. Дополнительные параметры обучения:

- Галочки для выбора дополнительных параметров обучения (список параметров будет указан в пункте 3.3.5).

8. Отображение общей стоимости курса:

- Поле, которое динамически рассчитывает и отображает общую стоимость на основе выбранных опций и продолжительности (см. пункт 3.3.4).

9. Кнопки действий:

- Кнопка «Отправить» для завершения заявки.
- Кнопка «Отмена» для закрытия модального окна без отправки.

### 3.3.4 Запрос расчета стоимости языковых курсов

Для динамического расчета стоимости обучения на языковом курсе используйте следующую формулу:

$$\begin{aligned} \text{Общая стоимость} &= ((\text{courseFeePerHour} \times \text{durationInHours} \times \\ \text{isWeekendOrHoliday}) &+ \text{morningSurcharge} + \text{eveningSurcharge}) \times \\ \text{studentsNumber} & \end{aligned}$$

Где:

- courseFeePerHour: стоимость обучения на курсе за один час.
- durationInHours: общая продолжительность курса, измеряется в часах.
- isWeekendOrHoliday: множитель, который корректирует стоимость, если курс проводится в выходные или праздничные дни.

- Будни (пн-пт): 1
- Выходные (сб, вс), праздники: 1.5
- morningSurcharge: Доплата за занятия, которые проводятся утром.
  - Для занятий с 9:00 до 12:00: 400 рублей
  - В остальное время: 0 рублей
- eveningSurcharge: Доплата за занятия, которые проходят вечером.
  - Для занятий с 18:00 до 20:00: 1000 рублей.
  - В остальное время: 0 руб.
- studentsNumber: кол-во студентов в заявке.

Submit an Application

**Course Enrollment Form** ×

Course Name:

Instructor's Name:

Start Date:  📅

Start Time:  ⌵

Duration (in weeks):

Number of Students:

Additional Options:

☐ Surcharge 1

☐ Surcharge 2

**Total Cost: 0 Rubles**

Рисунок 6: примерный макет формы зачисления на курс

### 3.3.5. Опции заказа

1. **Скидка за раннюю регистрацию (earlyRegistration):** Если студент регистрируется не менее чем за месяц вперед, стоимость снижается на 10% *(применяется автоматически, можно реализовать в виде чекбокса или появления какой-нибудь плашки в случае применения).*

2. **Скидка при групповой записи (groupEnrollment):** Студенты, зарегистрировавшиеся группой из 5 и более человек, получают скидку 15% от

общей стоимости курса (*применяется автоматически, можно реализовать в виде чекбокса или появления какой-нибудь плашки в случае применения*).

**3. Интенсивные курсы (intensiveCourse):** Выбор интенсивного курса (5 часов в неделю и более) увеличивает общую стоимость на 20% (*применяется автоматически, можно реализовать в виде чекбокса или появления какой-нибудь плашки в случае применения*).

**4. Дополнительные учебные материалы (supplementary):** Доступ к дополнительным онлайн-ресурсам и материалам в ходе курса увеличивает стоимость на 2000 рублей за каждого студента (*пользовательский выбор, чекбокс*).

**5. Индивидуальные занятия (personalized):** При выборе индивидуальных занятий в дополнение к регулярным занятиям общая стоимость увеличивается на 1500 рублей за каждую неделю курса (*пользовательский выбор, чекбокс*).

**6. Культурные экскурсии (excursions):** Участие в культурных экскурсиях, организованных курсом, увеличивает общую стоимость курса на 25% (*пользовательский выбор, чекбокс*).

**7. Оценка уровня владения языком (assessment):** Предварительная оценка уровня владения языком увеличивает общую стоимость курса на 300 рублей (*пользовательский выбор, чекбокс*).

**8. Доступ к интерактивной онлайн-платформе (interactive):** Доступ к передовой интерактивной платформе онлайн-обучения увеличивает стоимость курса на 50% (*пользовательский выбор, чекбокс*).

#### **Пояснения:**

**- Возможности скидок (1, 2):** Финансовые стимулы, поощряющие студентов записываться заранее или группами, что способствует увеличению числа участников.

- **Дополнительные услуги или опции (3-8):** Улучшения процесса обучения, повышающие его ценность, такие как интенсивные курсы для более быстрого обучения, дополнительные материалы для лучшего доступа к ресурсам и персонализированные сессии для индивидуального обучения.

## 4. Описание API

API для выполнения экзаменационного задания располагается по адресу: <http://exam-api-courses.std-900.ist.mospolytech.ru/>.

### 4.1 Модель данных

#### Курсы

Название параметра	Тип	Ограничение	Обязательное	Только для чтения	Примечание
id	int			Y	Устанавливается сервером
name	string	max 255 символов		Y	
description	string			Y	
teacher	string			Y	Имя преподавателя
level	string			Y	Уровень курса
total_length	int			Y	Продолжительность курса в неделях
week_length	int			Y	Количество часов в



					неделю
start_dates	Array[DateTime]			Y	Список доступных дат для начала курса
course_fee_per_hour	int			Y	Базовая стоимость обучения на курсе за час
created_at	DateTime			Y	Устанавливается сервером
updated_at	DateTime			Y	Устанавливается сервером

Пример в формате JSON:

```
{
    "id": 1,
    "name": "Introduction to Russian language",
    "description": "A beginner course on
Russian language learning.",
    "teacher": "Viktor Sergeevich",
    "level": "Beginner",
    "total_length": 8,
    "week_length": 2,
    "start_dates": ["2025-02-01T09:00:00",
"2025-02-01T12:00:00", "2025-02-01T17:00:00",
"2025-03-01T09:00:00", "2025-03-01T12:00:00",
"2025-03-01T17:00:00"],
    "course_fee_per_hour": 200,
    "created_at": "2025-01-05T17:30:00"
}
```

## Репетиторы

Название параметра	Тип	Ограничение	Обязательное	Только для чтения	Примечание
id	int			Y	Устанавливается сервером
name	string	max 255 символов		Y	
work_experience	int			Y	
languages_spoken	Array[String]			Y	
languages_offered	Array[String]			Y	
language_level	string			Y	
price_per_hour	int			Y	
created_at	DateTime			Y	Устанавливается сервером
updated_at	DateTime			Y	Устанавливается сервером

Пример в формате JSON:

```
{
  "id": 1,
  "name": "Irina Petrovna",
  "work_experience": 5,
  "languages_spoken": ["English", "Spanish",
"Russian"],
  "languages_offered": ["Russian", "English"],
  "language_level": "Advanced",
  "price_per_hour": 500
}
```

## Заявки

Название параметра	Тип	Ограничение	Обязательное	Только для чтения	Примечание
id	int			Y	
tutor_id	int		Y		Должно быть заполнено только одно из этих двух полей
course_id	int		Y		
date_start	date		Y		Передавать в формате YYYY-MM-DD
time_start	time		Y		Передавать в формате HH:MM
duration	int	Для репетиторов допустимые значения от 1 до 40	Y		Продолжительность в часах, для курсов вычисляется автоматически на основе данных о курсе
persons	int	от 1 до 20	Y		
price	int		Y		Общая стоимость
early_registration	bool		Y		Дополнительные опции, отсутствие их в теле запроса расценивается как false
group_enrollment	bool		Y		
intensive_course	bool		Y		
supplementary	bool		Y		
personalized	bool		Y		
excursions	bool		Y		
assessment	bool		Y		
interactive	bool		Y		

student_id	int			Y	id текущего пользователя, устанавливается сервером
created_at	DateTime			Y	устанавливается сервером
updated_at	DateTime			Y	устанавливается сервером

Пример в формате JSON:

```
{
    "id": 1,
    "tutor_id": 0,
    "course_id": 2,
    "date_start": "2025-01-15",
    "time_start": "14:00",
    "duration": 1,
    "persons": 2,
    "price": 2500,
    "early_registration": true,
    "group_enrollment": false,
    "intensive_course": true,
    "supplementary": true,
    "personalized": false,
    "excursions": false,
    "assessment": false,
    "interactive": true,
    "student_id": 1
}
```

## 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/json` (нужно вручную сериализовать данные и установить значение заголовка `Content-Type`).

### 4.2 Получить список курсов

URL	/api/courses
Method	GET
Response	JSON [{item1},{item2},...{itemN}]

#### 4.3 Получить список репетиторов

URL	/api/tutors
Method	GET
Response	JSON [{item1},{item2},...{itemN}]

#### 4.4 Получить список заявок

URL	/api/orders
Method	GET
Response	JSON [{item1},{item2},...{itemN}]

#### 4.5 Добавить заявку

URL	/api/orders
Method	POST
Params	field1=value1, field2=value2, ..... fieldN=valueN <b>Важно передать все обязательные поля</b>
Response	JSON {item}

#### 4.6 Редактировать заявку

URL	/api/orders/{order-id}
Method	PUT
Params	field1=value1, field2=value2,

	..... fieldN=valueN <b>Передавать можно только те поля, которые необходимо обновить (кроме boolean, их отсутствие в теле запроса изменит значения на false)</b>
Response	JSON {item}

#### 4.7 Удалить заявку

URL	/api/orders/{order-id}
Method	DELETE
Response	JSON {id: ID удаленной записи}

#### 4.8 Посмотреть заявку

URL	/api/orders/{order-id}
Method	GET
Response	JSON {item}

#### 4.9 Получить информацию о репетиторе

URL	api/tutors/{tutor-id}
Method	GET
Response	JSON {item}

#### 4.10 Получить информацию о курсе

URL	api/course/{course-id}
Method	GET

Response	JSON {item}
----------	-------------

## 5. Дополнительное задание

**Карта интерактивных учебных ресурсов:** Карта, показывающая, где студенты могут найти ресурсы (например, библиотеки, языковые кафе) для изучения иностранных языков.

**Карты:** Google Maps или Яндекс.Карты.

**Поиск ресурсов:** В поисковой строке должны быть релевантные термины, такие как:

- Языковой клуб
- Курсы иностранного языка
- Библиотеки с ресурсами на иностранных языках
- Культурные центры
- Кафе языкового обмена

**Фильтры результатов:** доступные параметры фильтрации на карте:

- Образовательные учреждения
- Общественные центры
- Публичные библиотеки
- Частные языковые курсы
- Языковые кафе или клубы

**Расположение отметок:** соответствующие ресурсы должны быть закреплены на карте.

**Подборка информации:**

Для каждого местоположения должна быть предоставлена важная информация, такая как:



- Название ресурса
- Адрес
- Часы работы
- Контактная информация (если имеется)
- Краткое описание предлагаемых услуг (например, занятия, разговорная практика, учебные материалы).

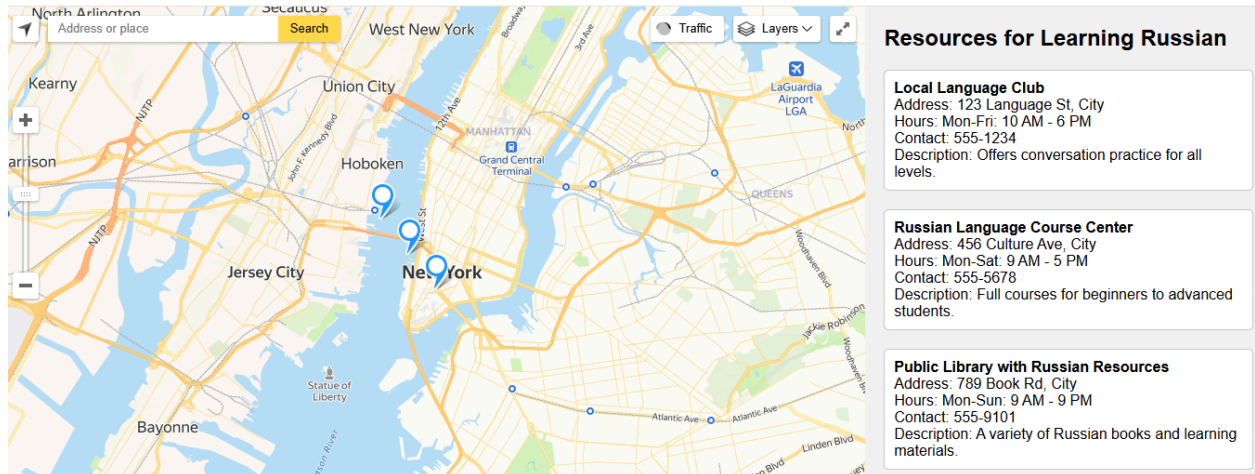


Рисунок 7: Пример визуализации карты учебных ресурсов

**Примечание:** не забудьте заменить `YOUR_API_KEY` в теге `<script>` вашего HTML на ваш фактический ключ API Яндекс.Карт.

```
<script
src="https://api-maps.yandex.ru/2.1/?apikey=YOUR_API_K
EY&lang=en_US" type="text/javascript" defer></script>
```

Чтобы получить API-ключ для Яндекс.Карт, необходимо авторизоваться на сайте разработчиков сервиса:

1. Перейдите по ссылке [developer.tech.yandex.ru](https://developer.tech.yandex.ru) и авторизуйтесь (введите логин и пароль) или зарегистрируйтесь.
2. Нажмите на кнопку «Получить ключ».

3. Выберите вариант «JavaScript API и HTTP Геокодер».

×

## Подключить API

Карты

☒ JavaScript API и HTTP Геокодер

☐ MapKit – мобильный SDK

☐ Матрица Расстояний и Построение Маршрута

☐ API Поиска по организациям

☐ Static API Яндекс.Карт

☐ Маршрутизация: Планирование и Мониторинг

☐ Маршрутизация: Календарное планирование

☐ Tiles API

☐ API Яндекс Локатор

☐ API Геосаджеста

Ещё

☐ API Яндекс.Расписаний

☐ API валидатора микроразметки

☐ API Поиска для сайта

☐ Safe Browsing API

☐ Яндекс.ОФД API

☐ API Кабинета Разработчика

☐ ADFOX

☐ API Яндекс.Коннекта

☐ Виджеты Яндекс.Доставки

☐ API платежей в приложении Яндекс

☐ Платформа турбо-сервисов

Продолжить

Отмена

4. Заполните пустые поля — обязательные позиции отмечены красной звёздочкой:

- а. ФИО, почта, телефон;
- б. отметьте, что ваш сервис отвечает условиям бесплатного использования версии;

\* Вы представляете юридическое лицо?

☐ Юридическое лицо

☒ Частное лицо

Для более быстрой обработки заявки, заполните, пожалуйста, поля ниже

Расскажите как хотите использовать API Карт

\* Цель использования

☒ Учебный проект

☐ Для разработки/тестирования

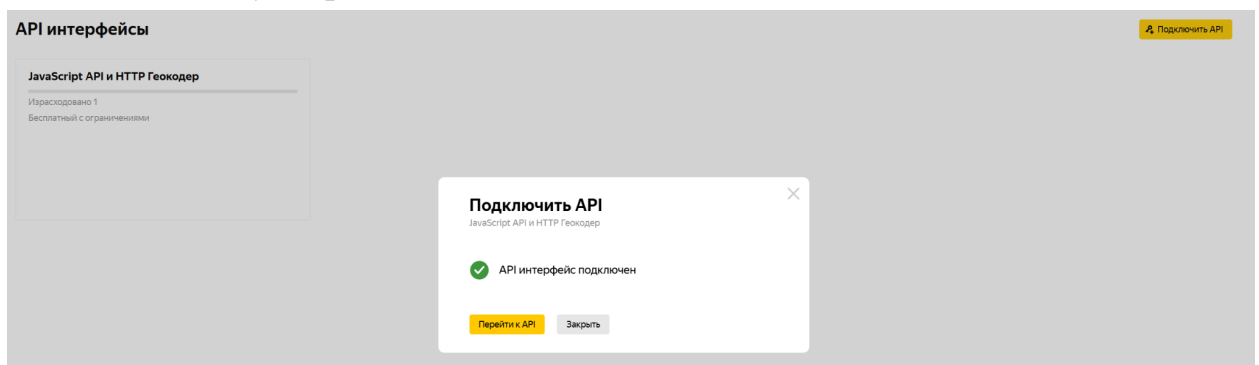
☐ НКО

☐ Нужен API ключ для конструктора карт

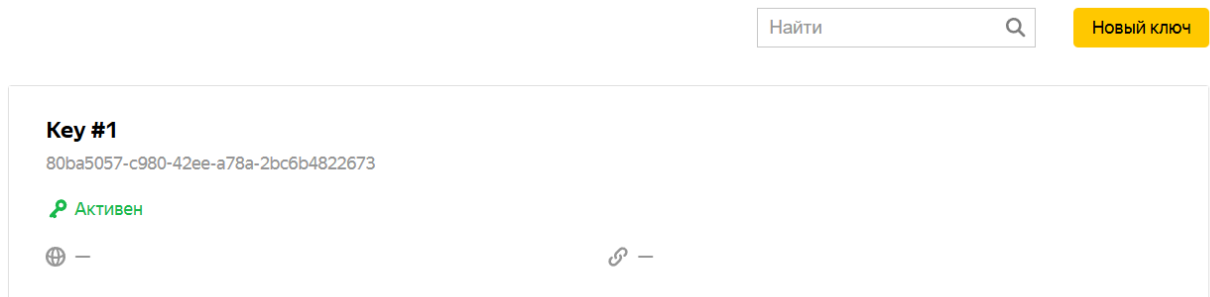
☐ Другая

Скорее всего, вам подходит бесплатное использование продуктов API Карт

- с. поставьте галочку в графе «Я принимаю условия».
5. Нажмите кнопку «Продолжить». API подключен.



6. Сервис переведёт вас на страницу Кабинета разработчика. Скопируйте ключ под заголовком «Ключи API».



## Приложение 1. Критерии оценки

- 1) Полностью сверстана главная страница и работает загрузка и отображение списка языковых курсов – 10 баллов (необходимо для сдачи проекта).
- 2) Оформление заявки (весь остальной функционал главной страницы) – до 10 баллов
- 3) Личный кабинет – до 10 баллов
- 4) Дополнительное задание – до 10 баллов

Максимальное количество баллов – 40.

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