

**МОСКОВСКАЯ ПРЕДПРОФЕССИОНАЛЬНАЯ  
ОЛИМПИАДА ШКОЛЬНИКОВ**  
**Профиль «Информационные технологии»**  
**Командный кейс №1 «Подготовка фотошаблона»**

---

## 1. Условия

Важной частью маршрута проектирования современных интегральных схем (ИС) в микроэлектронике является этап фотолитографии (рис. 1). Это процесс получения определенного рисунка на поверхности кремниевой пластины. Он начинается с нанесения на обрабатываемую поверхность тонкой фоточувствительной полимерной пленки (фоторезиста). Затем пленка засвечивается светом от специального источника через фотошаблон (маску) с заданным рисунком.

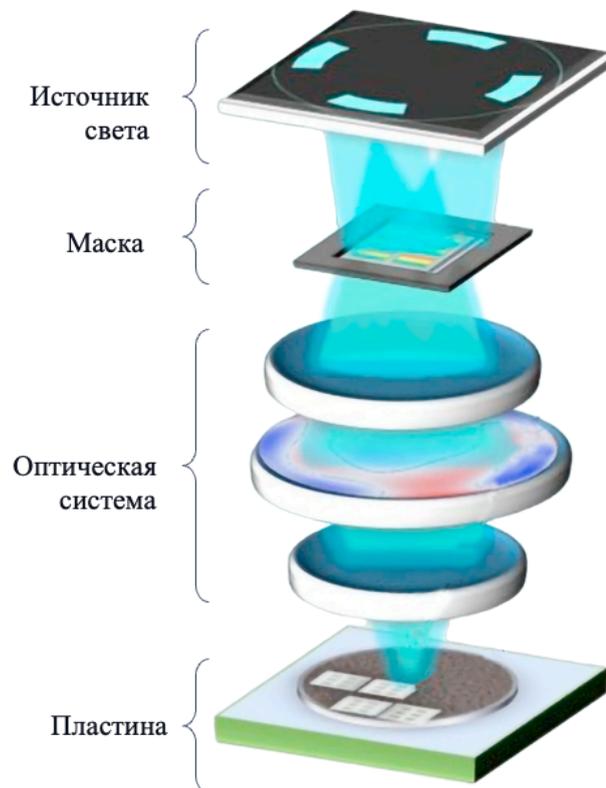


Рис. 1. Схема фотолитографии при производстве ИС

Ключевым фактором в указанном выше процессе является качество подготовки фотошаблона (рис. 2): из-за процессов дифракции и интерференции света структуры, проявленные на кремниевой подложке, искажаются. Чтобы этого избежать, фотошаблон модифицируется особым образом, приближая структуры к целевым. Такой процесс называется Optical Proximity Correction, или коррекция оптической близости.

МОСКОВСКАЯ ПРЕДПРОФЕССИОНАЛЬНАЯ  
ОЛИМПИАДА ШКОЛЬНИКОВ  
Профиль «Информационные технологии»  
Командный кейс №1 «Подготовка фотошаблона»

---



Рис. 2. Фотошаблон

На рисунках 3 и 4 схематично представлены результаты процесса фотолитографии с коррекцией оптической близости и без.



Рис. 3. Результат фотолитографии с коррекцией оптической близости



Рис. 4. Результат фотолитографии без коррекции оптической близости

Для подбора правильных алгоритмов модификации фотошаблона важно подготовить входные данные – структуры и их границы. Для решения этой задачи необходимо произвести высокоточную сегментацию (рис. 5) и выделение структур на общем фоне.

**МОСКОВСКАЯ ПРЕДПРОФЕССИОНАЛЬНАЯ  
ОЛИМПИАДА ШКОЛЬНИКОВ**  
Профиль «Информационные технологии»  
Командный кейс №1 «Подготовка фотошаблона»

---

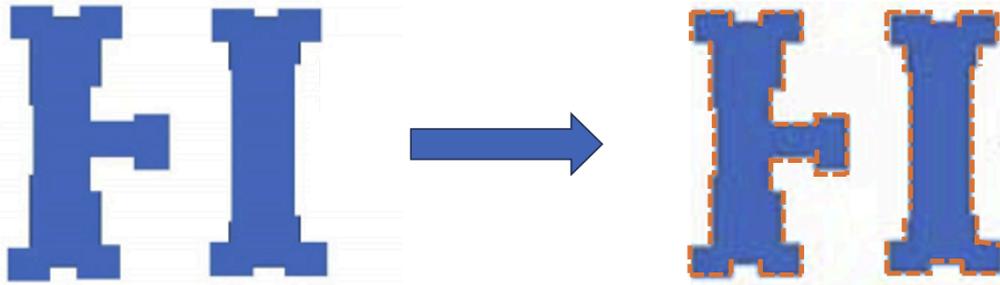


Рис. 5. Пример сегментации и выделения структур

## 2. Техническое задание

Разработать программное решение (приложение для Windows 10/Linux/macOS) сегментации, выделения и подсчета общего количества структур на фотошаблоне.

Функциональность программы:

- алгоритмы компьютерного зрения: сегментации, выделения и подсчета общего количества структур для формирования фотошаблона в формате изображения.

Входные данные:

- [фотошаблон](#) в формате JPG изображения 240x240 px.

Выходные данные:

- наложенные границы сегментов поверх входного изображения;
- CSV-файл с количеством выделенных сегментов на фотошаблоне (пример CSV-файла размещен по [ссылке](#)).

## 3. Рекомендации к выполнению

- использование системы контроля версий;
- ведение документации проекта (формат описания схем остается на усмотрение разработчиков);
- покрытие функционала модульным тестированием.

**МОСКОВСКАЯ ПРЕДПРОФЕССИОНАЛЬНАЯ  
ОЛИМПИАДА ШКОЛЬНИКОВ**  
**Профиль «Информационные технологии»**  
**Командный кейс №1 «Подготовка фотошаблона»**

---

#### **4. Требования к документации**

- Титульный лист (с указанием названия кейса и перечислением членов команды).
- Обоснование выбора языка программирования и используемых программных средств.
- Структурная и функциональная схемы программного продукта.
- Блок-схема работы основного алгоритма.
- Описание особенностей и аргументация выбранного типа СУБД.
- Схема базы данных.
- Программный код (ссылка на репозиторий), файл README должен включать:
  - краткое описание проекта;
  - инструкцию по установке/развертыванию;
  - ссылку на видеоролик.

#### **5. Требования к видеоролику**

- Видеоролик должен демонстрировать функционирование разработанного программного продукта в соответствии с регламентом испытаний.
- На видео или записи экрана необходимо продемонстрировать выполнение каждого испытания, описанного в регламенте, в соответствии с условиями.
- Видео должно однозначно подтверждать авторство участников (во время записи ролика необходимо четко произнести название команды, ФИО участников, номер школы, ФИО руководителя).
- Видеоролик необходимо разместить на стороннем видеохостинге («ВКонтакте», Rutube и др.)

**МОСКОВСКАЯ ПРЕДПРОФЕССИОНАЛЬНАЯ  
ОЛИМПИАДА ШКОЛЬНИКОВ**  
**Профиль «Информационные технологии»**  
**Командный кейс №1 «Подготовка фотошаблона»**

---

**6. Регламент испытаний**

1. Установка и запуск исполняемого файла приложения.
2. Выбор файла с фотошаблоном для дальнейшей сегментации.
3. Демонстрация визуализации результатов сегментирования для трех разных шаблонов.
4. Выгрузка результатов в выходное изображение и соответствующий CSV-файл для трех разных шаблонов.
5. Корректное завершение работы приложения на любых этапах работы.

**7. Методические материалы**

- <https://www.python.org>;
- <https://opencv.org/>;
- <https://pypi.org/project/opencv-python>;
- <https://github.com/AlexeyAB/darknet>;
- язык программирования Python;
- библиотека OpenCV для работы с компьютерным зрением;
- библиотека машинного обучения TensorFlow;
- детектор YOLO.

**МОСКОВСКАЯ ПРЕДПРОФЕССИОНАЛЬНАЯ  
ОЛИМПИАДА ШКОЛЬНИКОВ**  
**Профиль «Информационные технологии»**  
**Командный кейс №2 «SmartFridge: QR Control & IoT Sync»**

---

## **1. Условия**

Юные бизнесмены решили открыть свое собственное кафе, но столкнулись с проблемой: процесс инвентаризации продуктовых запасов в холодильнике занимал слишком много времени, а закупки продуктов становились неоптимальными, так как невозможно было уследить за актуальным содержимым. Руководством было принято решение разработать приложение для отслеживания содержимого в холодильнике и упрощения процесса ведения инвентаризации.

Участникам предлагается разработать приложение, которое будет отображать содержимое холодильника на основе данных, считанных с QR-кодов на продуктах, помещаемых в холодильник.

Приложение должно отображать подробную информацию о продуктах внутри холодильника, выводить аналитику потребления и уведомлять пользователя об истечении сроков годности продуктов.

## **2. Техническое задание**

Требуется составить набор QR-кодов для продуктов питания, содержащих информацию о них, а также приложение для управления содержимым холодильника со сканером QR-кодов (внешним или интегрированным в приложение).

**МОСКОВСКАЯ ПРЕДПРОФЕССИОНАЛЬНАЯ  
ОЛИМПИАДА ШКОЛЬНИКОВ**  
**Профиль «Информационные технологии»**  
**Командный кейс №2 «SmartFridge: QR Control & IoT Sync»**

---

*Функциональное задание:*

Обязательная для содержания в QR-кодах информация:

- название и тип продукта продукта, пример:
  - название - «Батон нарезной»;
  - тип - «Хлеб»;
- дата изготовления;
- дата истечения срока годности;
- масса/объем продукта и соответствующие единицы измерения в системе СИ;
- пищевая ценность продукта;
- тип измерения, например:
  - картофель - вес;
  - йогурт - штуки.

*Обязательная функциональность:*

- отображение на главной панели приложения полного списка содержимого холодильника и статусов срока годности продуктов;
- возможность удаления и добавления в холодильник новых продуктов путем сканирования QR-кода соответствующего продукта;
- функция поиска продуктов в холодильнике по названию (полному или частичному) и типу продукта;
- просмотр полной информации о конкретном продукте путем сканирования его QR-кода или выбора его в интерфейсе приложения;
- получение уведомлений о приближении даты истечения срока годности продукта в холодильнике;
- составление аналитики потребления по добавленным/удаленным из холодильника продуктам за указанный период времени;
- добавление и удаление продуктов по названию в «Список покупок» и просмотр этого списка.

*Требования к пользовательскому интерфейсу:*

В интерфейсе должны быть предусмотрены следующие экранные формы:

- главная панель, отображающая текущее содержимое холодильника с указанием количества каждого продукта и состояния срока годности

**МОСКОВСКАЯ ПРЕДПРОФЕССИОНАЛЬНАЯ  
ОЛИМПИАДА ШКОЛЬНИКОВ**  
**Профиль «Информационные технологии»**  
**Командный кейс №2 «SmartFridge: QR Control & IoT Sync»**

---

(например, приближается срок годности или продукт уже просрочен) и окно поиска;

- экран для сканирования QR-кодов (если не предусмотрен внешний сканер) с возможностью мгновенного добавления новых продуктов в систему;
- экран аналитики потребления, где пользователь может просматривать отчеты по потреблению продуктов за определенные периоды времени, настраиваемые пользователем;
- экран «Список покупок», содержащий названия и количество добавленных в него продуктов.

### **3. Рекомендации к выполнению**

- Приложение должно или работать на мобильных устройствах, или иметь веб-интерфейс.
- Устройство для считывания QR-кодов может быть любым, но рекомендуется использовать встроенную камеру мобильного телефона.
- В качестве примера структуры QR-кода рекомендуется обратить внимание на систему цифровой маркировки «Честный знак».
- Рекомендуется помечать специальными знаками в интерфейсе приложения продукты, содержащие в себе аллергены, лактозу и глютен.
- Необходимо сохранять информацию по продуктам в системе управления базами данных (СУБД). Выбор СУБД не регламентируется.
- Разработку рекомендуется вести с помощью системы контроля версий git.

### **4. Требования к документации**

- Титульный лист (с указанием названия кейса и перечислением членов команды).
- Обоснование выбора языка программирования и используемых программных средств.
- Структурная и функциональная схемы программного продукта.

**МОСКОВСКАЯ ПРЕДПРОФЕССИОНАЛЬНАЯ  
ОЛИМПИАДА ШКОЛЬНИКОВ**  
**Профиль «Информационные технологии»**  
**Командный кейс №2 «SmartFridge: QR Control & IoT Sync»**

---

- Блок-схема работы основного алгоритма.
- Описание особенностей и аргументация выбранного типа СУБД.
- Схема базы данных.
- Программный код (ссылка на репозиторий), файл README должен включать:
  - краткое описание проекта;
  - инструкцию по установке/развертыванию;
  - ссылку на видеоролик.

## **5. Требования к видеоролику**

- Видеоролик должен демонстрировать функционирование разработанного программного продукта в соответствии с регламентом испытаний.
- На видео или записи экрана необходимо продемонстрировать выполнение каждого испытания, описанного в регламенте, в соответствии с условиями.
- Видео должно однозначно подтверждать авторство участников (во время записи ролика необходимо четко произнести название команды, ФИО участников, номер школы, ФИО руководителя).
- Видеоролик необходимо разместить на стороннем видеохостинге («ВКонтакте», Rutube и др.)

**МОСКОВСКАЯ ПРЕДПРОФЕССИОНАЛЬНАЯ  
ОЛИМПИАДА ШКОЛЬНИКОВ**  
**Профиль «Информационные технологии»**  
**Командный кейс №2 «SmartFridge: QR Control & IoT Sync»**

---

## **6. Регламент испытаний**

1. Производится установка и запуск приложения.
2. Сканирование некоторого количества (от 2 до 4) QR-кодов, проверяется вывод информации по продуктам и возможность их добавления в холодильник.
3. Демонстрация интерфейса главной панели приложения с перечнем ранее добавленных в холодильник продуктов.
4. Демонстрация возможности удаления продуктов из холодильника.
5. Демонстрация работы поиска продуктов в содержимом холодильника.
6. Получение уведомлений о скором истечении срока годности продукта и просроченных продуктах (заранее необходимо подготовить QR-коды с необходимыми датами).
7. Демонстрация аналитики потребления продуктов за произвольный временной отрезок.
8. Демонстрация работы «Списка покупок».

## **7. Методические материалы**

- <https://www.python.org/>
- <https://sqlite.org/>
- <https://www.postgresql.org/>
- <https://flask.palletsprojects.com/en/2.0.x/>
- <https://developer.android.com/>
- <https://kotlinlang.org/docs/home.html>
- <https://docs.oracle.com/en/>
- <https://vuejs.org/>

# МОСКОВСКАЯ ПРЕДПРОФЕССИОНАЛЬНАЯ ОЛИМПИАДА ШКОЛЬНИКОВ

## Профиль «Информационные технологии»

### Командный кейс №3 «Распознавание и воспроизведение нот»

---

#### 1. Условия

В музыкальной нотации музыкальные звуки записываются с помощью набора символов, основные из которых — ноты. Ноты располагаются на линейке из пяти параллельных линий, называемой нотоносцем. Положение символа ноты на нотоносце определяет высоту обозначаемого звука и порядок исполнения звуков.

Участникам Олимпиады предлагается разработать приложение для распознавания нотного листа и его воспроизведения.

#### 2. Техническое задание

Приложение должно быть разработано на любом современном языке программирования, позволяющем создавать веб-приложения с использованием соответствующих фреймворков. Приложение должно иметь клиент-серверную архитектуру. Для работы приложения на стороне клиента требуется только веб-браузер. Пользователь должен иметь возможность сохранить аудиодорожки, преобразованные из нотных листов.

*Описание основной функциональности приложения.*

Регистрация и авторизация пользователей, для получения доступа в личный кабинет.

Загрузка в личном кабинете одного или нескольких нотных листов в виде изображений.

Распознавание нотных листов для формирования аудиодорожки в любом формате для аудиоданных. Если нотных листов несколько, то аудиодорожка должна быть склеена из этих листов.

Возможность скачивать преобразованные файлы с аудиоданными.

Сохранение аудиодорожки рекомендуется осуществлять на стороне сервера, для того чтобы пользователь мог обратиться к ранее преобразованным и сохраненным в личном кабинете.

**МОСКОВСКАЯ ПРЕДПРОФЕССИОНАЛЬНАЯ  
ОЛИМПИАДА ШКОЛЬНИКОВ  
Профиль «Информационные технологии»**

**Командный кейс №3 «Распознавание и воспроизведение нот»**

---

### **3. Рекомендации к выполнению**

- Изучите бесплатные сервисы по размещению готового веб-приложения (виртуальный хостинг).
- В случае затруднений с выбором фреймворка для веб-приложения рекомендуется рассмотреть Flask/Django (фреймворки для Python).
- Структурируйте код и программные модули по концепциям объектно-ориентированного программирования (ООП).
- Оформите все данные о проекте (название, описание, ссылка на испытания и пр.) в виде файла readme.md в репозитории проекта.

### **4. Требования к документации**

- Титульный лист (с указанием названия кейса и перечислением членов команды).
- Обоснование выбора языка программирования и используемых программных средств.
- Структурная и функциональная схемы программного продукта.
- Блок-схема работы основного алгоритма.
- Описание особенностей и аргументация выбранного типа СУБД.
- Схема базы данных.
- Программный код (ссылка на репозиторий), файл README должен включать:
  - краткое описание проекта;
  - инструкцию по установке/развертыванию;
  - ссылку на видеоролик.

### **5. Требования к видеоролику**

- Видеоролик должен демонстрировать функционирование разработанного программного продукта в соответствии с регламентом испытаний.
- На видео или записи экрана необходимо продемонстрировать выполнение каждого испытания, описанного в регламенте, в соответствии с условиями.

**МОСКОВСКАЯ ПРЕДПРОФЕССИОНАЛЬНАЯ  
ОЛИМПИАДА ШКОЛЬНИКОВ  
Профиль «Информационные технологии»**

**Командный кейс №3 «Распознавание и воспроизведение нот»**

---

- Видео должно однозначно подтверждать авторство участников (во время записи ролика необходимо четко произнести название команды, ФИО участников, номер школы, ФИО руководителя).
- Видеоролик необходимо разместить на стороннем видеохостинге («ВКонтакте», Rutube и др.)

## **6. Регламент испытаний**

1. Загрузка нотных листов для преобразования.
2. Демонстрация того, что ноты на листах распознаны.
3. Скачивание аудиодорожки.
4. Демонстрация того, что распознанные листы хранятся в личном кабинете пользователя.
5. Удаление ранее распознанных аудиодорожек.

## **7. Методические материалы**

- Документация по инструментам Python <https://www.python.org/>
- Документация по инструментам Node.js <https://nodejs.org/>
- Документация по инструментам SQLite <https://sqlite.org/>
- Документация по инструментам React.js <https://reactjs.org/>
- Документация по инструментам Mido <https://mido.readthedocs.io/en/stable/>
- Документация по инструментам PyTorch <https://pytorch.org/>
- Документация по инструментам Keras <https://keras.io/>

**МОСКОВСКАЯ ПРЕДПРОФЕССИОНАЛЬНАЯ  
ОЛИМПИАДА ШКОЛЬНИКОВ**  
**Профиль «Информационные технологии»**  
**Командный кейс №4 «Антифишинговый плагин для браузера»**

---

## **1. Введение**

В последние десятилетия интернет стал неотъемлемой частью жизни миллиардов людей по всему миру. Мы ежедневно используем его для работы, общения, покупок и других задач, что привело к появлению новых угроз безопасности. Одной из самых опасных является фишинг — тип кибератаки, при которой злоумышленники маскируются под надежные источники, чтобы обманом получить конфиденциальные данные, такие как пароли и данные банковских счетов. Фишинг часто реализуется через фальшивые веб-сайты или электронные письма, которые кажутся легитимными. Неопытные пользователи могут стать жертвами таких атак, что приводит к кражам данных и финансовым потерям.

Несмотря на существующие меры безопасности, включая двухфакторную аутентификацию, фишинговые атаки продолжают оставаться эффективными, затрагивая миллионы пользователей ежедневно. Это подчеркивает необходимость разработки новых решений для их защиты. Одним из таких решений может стать антифишинговый плагин для браузера, способный анализировать веб-страницы, выявлять потенциально опасные ресурсы и блокировать их. Это существенно повысит уровень безопасности пользователей в сети.

Разработка антифишингового плагина требует создания алгоритмов, способных точно распознавать фишинговые сайты, минимизируя ложные срабатывания. Также важно обеспечить совместимость плагина с различными браузерами и операционными системами и сделать его простым в установке и использовании. Антифишинговый плагин может

**МОСКОВСКАЯ ПРЕДПРОФЕССИОНАЛЬНАЯ  
ОЛИМПИАДА ШКОЛЬНИКОВ**  
**Профиль «Информационные технологии»**  
**Командный кейс №4 «Антифишинговый плагин для браузера»**

---

значительно снизить количество успешных фишинговых атак, защищая пользователей от кражи данных. Этот проект не только предлагает техническое решение, но и способствует повышению осведомленности пользователей о киберугрозах. Основная цель проекта — создание эффективного инструмента для защиты от фишинга, который поможет сделать интернет безопаснее.

## **2. Условия**

### **1. Совместимость с Google Chrome:**

- плагин должен быть разработан с учетом требований Google Chrome, используя Chrome Extension API, и совместим с последними стабильными версиями браузера.

### **2. Распознавание фишинговых сайтов:**

- плагин должен в реальном времени анализировать URL и содержимое веб-страниц для определения потенциально фишинговых сайтов, учитывая методы маскировки (подмена доменов, похожие символы, фишинговые формы).

### **3. Интеграция с браузером:**

- плагин должен интегрироваться с интерфейсом Google Chrome, предоставляя пользователю интуитивно понятный интерфейс для управления. Интерфейс должен включать иконку в панели инструментов и отображать уведомления при обнаружении фишинговых сайтов.

### **4. Эффективность и точность:**

**МОСКОВСКАЯ ПРЕДПРОФЕССИОНАЛЬНАЯ  
ОЛИМПИАДА ШКОЛЬНИКОВ**  
**Профиль «Информационные технологии»**  
**Командный кейс №4 «Антифишинговый плагин для браузера»**

---

- плагин должен обеспечивать высокую точность распознавания фишинговых сайтов с минимальными ложными срабатываниями, не влияя на производительность браузера.

5. Безопасность и конфиденциальность:

- плагин должен соответствовать строгим стандартам безопасности, предотвращать утечки данных и быть прозрачным в отношении сбора и использования данных, позволяя пользователям управлять настройками конфиденциальности.

6. Удобство использования:

- плагин должен быть простым в установке и использовании, с интуитивно понятным интерфейсом, содержащим четкие инструкции и подсказки. Включение раздела справки или FAQ будет преимуществом.

7. Тестирование и качество:

- плагин должен пройти тщательное тестирование на всех этапах разработки, включая функциональное и производительное тестирование, с особым вниманием к совместимости с различными версиями Google Chrome и другими расширениями.

**МОСКОВСКАЯ ПРЕДПРОФЕССИОНАЛЬНАЯ  
ОЛИМПИАДА ШКОЛЬНИКОВ**  
**Профиль «Информационные технологии»**  
**Командный кейс №4 «Антифишинговый плагин для браузера»**

---

### 3. Техническое задание

*Функциональное задание:*

Функциональное задание описывает основные и дополнительные функции, которые должен выполнять разрабатываемый антифишинговый плагин для браузера Google Chrome. Задание также включает требования к пользовательскому интерфейсу, чтобы обеспечить удобство и эффективность использования плагина.

*Обязательная функциональность:*

#### 1. Распознавание фишинговых сайтов.

- Плагин должен проверять URL и содержимое загружаемых страниц на наличие признаков фишинга, используя заранее определенные списки и правила.
- Источники данных для проверки: плагин использует обновляемые списки URL, классифицированные как фишинговые, например: OpenPhish, PhishTank или Google Safe Browsing API.
- Правила и эвристики: в дополнение к спискам плагин должен иметь набор правил и шаблонов для выявления подозрительных признаков, таких как структура URL, необычные элементы HTML или JavaScript, характерные для фишинговых сайтов.

#### 2. Блокировка фишинговых сайтов.

**МОСКОВСКАЯ ПРЕДПРОФЕССИОНАЛЬНАЯ  
ОЛИМПИАДА ШКОЛЬНИКОВ**  
**Профиль «Информационные технологии»**  
**Командный кейс №4 «Антифишинговый плагин для браузера»**

---

- Реализация блокировки до полной загрузки страницы: при обнаружении фишинговой угрозы плагин должен предотвращать загрузку содержимого страницы, чтобы снизить риск взаимодействия пользователя с вредоносным сайтом. Блокировка должна выполняться с использованием Chrome Extension API, особенно API `webRequest`, чтобы вмешиваться в процесс загрузки до его завершения.
- Оповещение пользователя: в случае блокировки пользователю должно выводиться уведомление о потенциальной угрозе с объяснением причин блокировки.

### 3. Управление белыми списками.

- Интерфейс управления белыми списками: плагин должен предоставлять возможность пользователю добавлять домены в белый список. URL из белого списка не будут проходить проверку и блокировку.
- Проверка белых списков: при обращении к сайту плагин проверяет, находится ли домен в белом списке, и при необходимости исключает его из блокировки. Белый список должен храниться локально для быстрого доступа и минимизации задержек.

### 4. История блокировок.

- Плагин должен сохранять историю заблокированных сайтов, включая URL, дату блокировки и причину блокировки.
- Пользователь должен иметь возможность просматривать и фильтровать историю блокировок.

**МОСКОВСКАЯ ПРЕДПРОФЕССИОНАЛЬНАЯ  
ОЛИМПИАДА ШКОЛЬНИКОВ**  
**Профиль «Информационные технологии»**  
**Командный кейс №4 «Антифишинговый плагин для браузера»**

---

5. Настройки конфиденциальности.

- Плагин должен позволять пользователю управлять сбором статистики, отправкой отчетов, и возможностью удаления истории блокировок.

*Дополнительная функциональность:*

1. Настраиваемые уровни фильтрации.

- Пользователь должен иметь возможность настраивать уровень строгости фильтрации с помощью параметров, таких как ключевые слова, черные списки доменов и другие правила.

2. Выгрузка отчетов.

- Плагин должен предоставлять возможность выгрузки отчетов о проведенных блокировках в формате CSV или JSON.

3. Интеграция с внешними сервисами.

- Возможность интеграции плагина с внешними сервисами для проверки URL на предмет фишинговых угроз.

*Требования к пользовательскому интерфейсу:*

1. Настраиваемые уровни фильтрации.

- Пользователь должен иметь возможность настраивать уровень строгости фильтрации с помощью параметров, таких как ключевые слова, черные списки доменов и другие правила.

2. Выгрузка отчетов.

- Плагин должен предоставлять возможность выгрузки отчетов о проведенных блокировках в формате CSV или JSON.

3. Интеграция с внешними сервисами.

**МОСКОВСКАЯ ПРЕДПРОФЕССИОНАЛЬНАЯ  
ОЛИМПИАДА ШКОЛЬНИКОВ**  
**Профиль «Информационные технологии»**  
**Командный кейс №4 «Антифишинговый плагин для браузера»**

---

- Поддерживаемые внешние сервисы: интеграция с API популярных антифишинговых сервисов, таких как Google Safe Browsing, PhishTank и OpenPhish, позволяет использовать актуальные данные для обнаружения угроз. Эти сервисы предоставляют обновляемые списки фишинговых URL и улучшают точность проверки.
- Архитектурные особенности.
  - Асинхронные запросы и кеширование: чтобы минимизировать задержки, плагин должен использовать асинхронные запросы и кратковременное кеширование результатов проверок.
  - Очередь запросов и лимиты API: для снижения нагрузки и предотвращения блокировок из-за лимитов API важно настроить очередь запросов и тайм-ауты, что обеспечит устойчивую работу.
- Функциональность.
  - Обновляемые данные: внешние сервисы обеспечивают актуальные данные для предотвращения фишинга, однако могут вызывать сетевые задержки.
  - Резервные проверки: в случае недоступности внешних API плагин может использовать локальные списки, чтобы не терять функциональность.

**МОСКОВСКАЯ ПРЕДПРОФЕССИОНАЛЬНАЯ  
ОЛИМПИАДА ШКОЛЬНИКОВ**  
**Профиль «Информационные технологии»**  
**Командный кейс №4 «Антифишинговый плагин для браузера»**

---

#### **4. Рекомендации к выполнению**

##### 1. Исследование и анализ угроз.

- Анализ решений: изучите текущие антифишинговые плагины и методы, определите их сильные и слабые стороны для разработки более эффективного решения.
- Обновление базы данных: регулярно обновляйте базы данных фишинговых сайтов и черные списки для актуальности.

##### 2. Проектирование и архитектура.

- Модульная архитектура: используйте модульный подход для упрощения обслуживания и добавления новых функций.
- Пользовательский интерфейс: разработайте прототипы интерфейса, протестируйте их с пользователями, обеспечивая удобство и интуитивность. Используйте принципы UX/UI.

##### 3. Разработка и тестирование.

- Кодирование: следуйте лучшим практикам программирования и соблюдайте стиль кода для улучшения читаемости и поддержки.
- Тестирование: проводите функциональное и нефункциональное тестирование, включая тестирование производительности и безопасности. Используйте автоматизированные тесты.
- Совместимость: убедитесь, что плагин корректно работает с разными версиями Google Chrome и на различных операционных системах.

##### 4. Безопасность и конфиденциальность.

**МОСКОВСКАЯ ПРЕДПРОФЕССИОНАЛЬНАЯ  
ОЛИМПИАДА ШКОЛЬНИКОВ**  
**Профиль «Информационные технологии»**  
**Командный кейс №4 «Антифишинговый плагин для браузера»**

---

- Безопасность данных: применяйте надежные методы защиты данных и шифрования для предотвращения утечек.

5. Интерфейс и удобство использования.

- Простота и ясность: сделайте интерфейс простым и понятным, чтобы пользователь легко находил и использовал все функции плагина.

**5. Требования к документации**

- Титульный лист (с указанием названия кейса и перечислением членов команды).
- Обоснование выбора языка программирования и используемых программных средств.
- Структурная и функциональная схемы программного продукта.
- Блок-схема работы основного алгоритма.
- Описание особенностей и аргументация выбранного типа СУБД.
- Схема базы данных.
- Программный код (ссылка на репозиторий), файл README должен включать:
  - краткое описание проекта;
  - инструкцию по установке/развертыванию;
  - ссылку на видеоролик.

**6. Требования к видеоролику**

- Видеоролик должен демонстрировать функционирование разработанного программного продукта в соответствии с регламентом испытаний.
- На видео или записи экрана необходимо продемонстрировать выполнение каждого испытания, описанного в регламенте, в соответствии с условиями.

**МОСКОВСКАЯ ПРЕДПРОФЕССИОНАЛЬНАЯ  
ОЛИМПИАДА ШКОЛЬНИКОВ**

**Профиль «Информационные технологии»**

**Командный кейс №4 «Антифишинговый плагин для браузера»**

---

- Видео должно однозначно подтверждать авторство участников (во время записи ролика необходимо четко произнести название команды, ФИО участников, номер школы, ФИО руководителя).
- Видеоролик необходимо разместить на стороннем видеохостинге («ВКонтакте», Rutube и др.)

**МОСКОВСКАЯ ПРЕДПРОФЕССИОНАЛЬНАЯ  
ОЛИМПИАДА ШКОЛЬНИКОВ**  
**Профиль «Информационные технологии»**  
**Командный кейс №4 «Антифишинговый плагин для браузера»**

---

## **7. Регламент испытаний**

Этапы проведения испытаний

### 1.1. Подготовка к испытаниям.

- Установка плагина: установить антифишинговый плагин на последнюю стабильную версию браузера Google Chrome на нескольких тестовых устройствах с различными операционными системами (Windows, macOS, Linux).
- Настройка окружения: обеспечить доступ к интернету и подготовить тестовые аккаунты, если необходимо. Создать список тестовых URL, включающий как известные фишинговые сайты, так и легитимные сайты.
- Инструменты мониторинга: настроить инструменты для мониторинга производительности и использования ресурсов плагином (например, Chrome Developer Tools, Task Manager).

### 2. Функциональное тестирование.

#### 2.1. Распознавание и блокировка фишинговых сайтов.

- Тест 1: Блокировка известных фишинговых сайтов.
  - Откройте в браузере сайт, известный как фишинговый, и убедитесь, что плагин успешно распознает его и блокирует доступ.
  - Зафиксируйте уведомление плагина, которое должно содержать информацию о причине блокировки и опции для пользователя (например, «Подробнее», «Исключить из блокировки»).
- Тест 2: Распознавание нового фишингового сайта.

**МОСКОВСКАЯ ПРЕДПРОФЕССИОНАЛЬНАЯ  
ОЛИМПИАДА ШКОЛЬНИКОВ**  
**Профиль «Информационные технологии»**  
**Командный кейс №4 «Антифишинговый плагин для браузера»**

---

- Создайте или выберите новый фишинговый сайт, который еще не был добавлен в черные списки, и откройте его в браузере.
- Проверьте, как плагин распознает и блокирует этот сайт. Зафиксируйте результат.

## 2.2. Работа с белым списком сайтов.

- Тест 3: Добавление сайта в белый список.
  - Добавьте безопасный сайт, который плагин ошибочно определяет как фишинговый, в белый список.
  - Откройте сайт снова и убедитесь, что плагин не блокирует доступ к этому сайту.
- Тест 4: Удаление сайта из белого списка.
  - Удалите сайт из белого списка.
  - Откройте сайт снова и проверьте, что плагин снова проверяет сайт и блокирует его, если он подозрительный.

## 2.3. Тестирование производительности.

- Тест 5: Влияние плагина на время загрузки страниц.
  - Замерьте время загрузки нескольких сайтов с включенным и отключенным плагином.
  - Сравните результаты и оцените, насколько плагин влияет на производительность браузера.

## 3. Тестирование стабильности и надежности.

### 3.1. Поведение после сбоя.

- Тест 8: Восстановление после сбоя.

**МОСКОВСКАЯ ПРЕДПРОФЕССИОНАЛЬНАЯ  
ОЛИМПИАДА ШКОЛЬНИКОВ**  
**Профиль «Информационные технологии»**  
**Командный кейс №4 «Антифишинговый плагин для браузера»**

---

- Принудительно завершите работу браузера и затем откройте его снова.
- Проверьте, сохраняются ли настройки плагина, и продолжает ли он корректно выполнять свои функции после перезапуска браузера.

#### 4. Тестирование пользовательского интерфейса.

##### 4.1. Интерфейс и удобство использования.

- Тест 9: Оценка удобства интерфейса.
  - Проанализируйте, насколько интуитивно понятен и удобен интерфейс плагина для пользователя.
  - Проверьте доступность всех основных функций, таких как управление белым списком, настройка уровней защиты и просмотр истории блокировок.

##### 4.2. Работа уведомлений.

- Типы уведомлений:
  - Оповещение о блокировке фишингового сайта: при попытке пользователя открыть фишинговый сайт плагин блокирует страницу и выводит уведомление.
  - Информация о причине блокировки: уведомление должно содержать причину блокировки (например, ссылка определена как фишинговая по данным конкретного сервиса или правилам плагина).
- Тест 10: Проверка системы уведомлений.
  - Создайте условия для генерации уведомлений плагина (например, при попытке посетить фишинговый сайт).

**МОСКОВСКАЯ ПРЕДПРОФЕССИОНАЛЬНАЯ  
ОЛИМПИАДА ШКОЛЬНИКОВ**  
**Профиль «Информационные технологии»**  
**Командный кейс №4 «Антифишинговый плагин для браузера»**

---

- Убедитесь, что уведомления отображаются своевременно, корректно и содержат необходимую информацию для пользователя.

## 8. Методические материалы

1. [Python](#) – официальный сайт языка программирования Python. Необходим для разработки вспомогательных инструментов и скриптов, а также для тестирования.
2. [PyCharm](#) – интегрированная среда разработки (IDE) для Python. Предоставляет мощные инструменты для редактирования, отладки и тестирования кода.
3. [Selenium WebDriver](#) – инструмент для автоматизации веб-браузеров, который может быть использован для тестирования работы плагина в браузере Google Chrome.
4. **Flask** – легковесный веб-фреймворк для Python, который может быть использован для разработки серверной части, если требуется взаимодействие плагина с сервером.
5. [pip](#) – официальный менеджер пакетов Python, который используется для установки и управления библиотеками и зависимостями проекта.
6. [Git](#) – система контроля версий, которая позволяет отслеживать изменения в коде и управлять проектом совместно с командой.

**МОСКОВСКАЯ ПРЕДПРОФЕССИОНАЛЬНАЯ  
ОЛИМПИАДА ШКОЛЬНИКОВ**  
**Профиль «Информационные технологии»**  
**Командный кейс №4 «Антифишинговый плагин для браузера»**

---

7. [GitHub](#) – платформа для хостинга Git-репозитория, предоставляющая возможности для управления проектами, совместной разработки и публикации исходного кода.
8. [Jupyter Notebook](#) – интерактивная среда для разработки и выполнения кода Python, особенно удобная для прототипирования и анализа данных.
9. [Sphinx](#) – инструмент для создания документации из исходного кода Python. Поддерживает генерацию документации в различных форматах.
10. [pytest](#) – фреймворк для тестирования Python, который поддерживает как модульное, так и интеграционное тестирование.
11. Chrome Developer Tools – встроенные в Google Chrome инструменты разработчика, которые помогают отлаживать и тестировать расширения и веб-приложения.
12. [Anaconda](#) – дистрибутив Python, который включает в себя множество научных библиотек и инструментов для анализа данных и машинного обучения, что может быть полезно при разработке сложных алгоритмов для плагина.

**МОСКОВСКАЯ ПРЕДПРОФЕССИОНАЛЬНАЯ  
ОЛИМПИАДА ШКОЛЬНИКОВ**  
**Профиль «Информационные технологии»**  
**Командный кейс №5 «Управление спортивным инвентарем»**

---



## **1. Условия**

Эффективный учет и контроль спортивного инвентаря в школе играет важную роль в обеспечении высокого уровня проведения спортивных мероприятий и занятий физической культурой. Необходимо разработать приложение, которое позволит отслеживать наличие, состояние и распределение спортивного инвентаря, а также планировать его закупку.

## **2. Техническое задание**

*Функциональное задание:*

Требуется разработать приложение с web-интерфейсом для учета школьного спортивного инвентаря. Приложение должно предоставлять различные уровни доступа для пользователей и администраторов, позволять вести учет инвентаря, распределять его среди пользователей, а также планировать закупки.

Обязательная функциональность:

Необходимо предусмотреть роли с уровнями доступа:

- администратор;
- пользователь.

Обязательная функциональность приложения для администратора:

- авторизация администратора в приложении;
- добавление позиций инвентаря с указанием названия, количества;
- редактирование позиций инвентаря в части состояния (новый, используемый, сломанный), количества, названия;

**МОСКОВСКАЯ ПРЕДПРОФЕССИОНАЛЬНАЯ  
ОЛИМПИАДА ШКОЛЬНИКОВ**  
**Профиль «Информационные технологии»**  
**Командный кейс №5 «Управление спортивным инвентарем»**

---

- закрепление за пользователями инвентаря;
- планирование и управление закупками инвентаря: добавление в план закупок с указанием цены и планируемого поставщика (учитывается только название поставщика);
- создание отчётов по использованию и состоянию инвентаря.

Обязательная функциональность приложения для пользователя:

- регистрация и авторизация пользователя;
- просмотр доступного инвентаря и его состояния;
- создание заявок на получение инвентаря;
- отслеживание статуса заявок на получение инвентаря.

Дополнительная функциональность приложения со стороны пользователя:

- создание заявки о необходимости ремонта или замены инвентаря.

Дополнительная функциональность приложения со стороны администратора:

- поддержка интеграции с внешними системами для автоматизации закупок (выбор системы не регламентируется).

### **3. Рекомендации к выполнению**

- Рекомендуется использовать систему управления базами данных (СУБД) для хранения данных. Выбор СУБД не регламентируется.
- Рекомендуется использовать клиент-серверную архитектуру.
- Рекомендуется использовать систему контроля версий для ведения проекта.
- Рекомендуется предусмотреть автоматизированную установку приложения на целевой машине.

### **4. Требования к документации**

- Титульный лист (с указанием названия кейса и перечислением членов команды).
- Обоснование выбора языка программирования и используемых программных средств.

**МОСКОВСКАЯ ПРЕДПРОФЕССИОНАЛЬНАЯ  
ОЛИМПИАДА ШКОЛЬНИКОВ**  
**Профиль «Информационные технологии»**  
**Командный кейс №5 «Управление спортивным инвентарем»**

---

- Структурная и функциональная схемы программного продукта.
- Блок-схема работы основного алгоритма.
- Описание особенностей и аргументация выбранного типа СУБД.
- Схема базы данных.
- Программный код (ссылка на репозиторий), файл README должен включать:
  - краткое описание проекта;
  - инструкцию по установке/развертыванию;
  - ссылку на видеоролик.

**МОСКОВСКАЯ ПРЕДПРОФЕССИОНАЛЬНАЯ  
ОЛИМПИАДА ШКОЛЬНИКОВ**  
**Профиль «Информационные технологии»**  
**Командный кейс №5 «Управление спортивным инвентарем»**

---

## **5. Требования к видеоролику**

- Видеоролик должен демонстрировать функционирование разработанного программного продукта в соответствии с регламентом испытаний.
- На видео или записи экрана необходимо продемонстрировать выполнение каждого испытания, описанного в регламенте, в соответствии с условиями.
- Видео должно однозначно подтверждать авторство участников (во время записи ролика необходимо четко произнести название команды, ФИО участников, номер школы, ФИО руководителя).
- Видеоролик необходимо разместить на стороннем видеохостинге («ВКонтакте», Rutube и др.)

## **6. Регламент испытаний**

Испытания должны включать проверку всех функциональных требований, указанных в техническом задании, в том числе:

- регистрация и авторизация пользователей;
- добавление не менее четырех позиций инвентаря, редактирование не менее двух позиций, удаление одной позиции администратором;
- назначение не менее двух позиций инвентаря двум разным пользователям;
- добавление не менее двух позиций в план закупок администратором;
- оформление двух заявок на новый инвентарь пользователями, одобрение одной заявки и отклонение одной заявки администратором;
- формирование отчета администратором об использовании инвентаря с указанием статуса и пользователя;

**МОСКОВСКАЯ ПРЕДПРОФЕССИОНАЛЬНАЯ  
ОЛИМПИАДА ШКОЛЬНИКОВ**  
**Профиль «Информационные технологии»**  
**Командный кейс №5 «Управление спортивным инвентарем»**

---

- тестирование обработки исключительных ситуаций (например, недостаток инвентаря для выполнения заявки).

## 7. Методические материалы

Для успешного выполнения задачи рекомендуется ознакомиться с материалами:

- [введение в HTML, CSS и JavaScript](#);
- использование фреймворков для создания веб-приложений: [Flask](#), [Django](#), [Spring](#);
- работа с базами данных (например, [SQLite](#), [MySQL](#));
- основы управления проектами, например [git](#).

**МОСКОВСКАЯ ПРЕДПРОФЕССИОНАЛЬНАЯ  
ОЛИМПИАДА ШКОЛЬНИКОВ**  
**Профиль «Информационные технологии»**  
**Командный кейс №6 «Образовательный портал»**

---

## **1. Условия**

В современном мире школьники 9-11 классов сталкиваются с возрастающими требованиями к уровню знаний и навыков, необходимых для успешной сдачи экзаменов и поступления в вузы. Несмотря на наличие большого количества образовательных ресурсов, многие из них недостаточно структурированы, зачастую в них отсутствует персонализированный подход и интерактивность, что усложняет процесс подготовки и снижает его эффективность. Школьники испытывают информационную перегрузку, недостаток поддержки и мотивации, они вынуждены использовать множество различных образовательных платформ, что затрудняет их учебу.

Для решения этой проблемы предлагается разработать приложение с веб-интерфейсом - образовательный портал, который объединит все необходимые учебные материалы, интерактивные задания и тесты в одной удобной среде. Портал должен обеспечивать персонализированный подход к обучению, адаптироваться под уровень подготовки и интересы каждого ученика, предоставлять инструменты для отслеживания прогресса и анализа успеваемости, а также мотивировать учащихся с помощью геймификации и других интерактивных элементов. Реализация этого проекта позволит создать уникальную образовательную платформу, которая поможет школьникам 9-11 классов эффективно подготовиться к экзаменам, повысить уровень знаний и развить необходимые навыки для дальнейшего обучения и профессионального роста.

## **2. Техническое задание**

*Функциональное задание:*

Обязательная функциональность:

- регистрация и авторизация пользователей (профили ученика, учителя и администратора с ролями разного уровня доступа), реализация механизма с помощью JWT, OAuth2.0 (вход через сторонние сервисы Yandex, Google);

**МОСКОВСКАЯ ПРЕДПРОФЕССИОНАЛЬНАЯ  
ОЛИМПИАДА ШКОЛЬНИКОВ**  
**Профиль «Информационные технологии»**  
**Командный кейс №6 «Образовательный портал»**

---

- модуль для управления курсами, содержащими структурированные учебные материалы:
  - лекции - в виде презентаций;
  - видеоуроки - в виде видеозаписей;
  - методические материалы - в виде текстовых документов;
- использование системы тегов и категорий для удобной навигации;
- поддержка интерактивных элементов (встроенные тесты, задания);
- модуль подготовки к экзаменам (тестовое решение экзаменационных вариантов);
- статистика и успеваемость;
- обсуждения (форум).

*Требования к пользовательскому интерфейсу:*

- использование компонентного подхода для создания динамичного и адаптивного интерфейса;
- обеспечение кроссплатформенности;
- производительность:
  - интерфейс должен быстро реагировать на действия пользователя, обеспечивая минимальное время отклика;
  - минимизация количества запросов к серверу и сокращение объема передаваемых данных, например, за счет использования кеша и сжатия;
  - использование техники «ленивой загрузки» для асинхронной загрузки несущественных компонентов и ресурсов, для ускорения первоначальной загрузки страницы.

### **3. Рекомендации к выполнению**

- разделите проект на четкие этапы (планирование, дизайн, разработка, тестирование, развертывание) и придерживайтесь установленного графика;
- поддерживайте постоянную связь с членами команды и заинтересованными сторонами, чтобы оперативно решать возникающие вопросы;

**МОСКОВСКАЯ ПРЕДПРОФЕССИОНАЛЬНАЯ  
ОЛИМПИАДА ШКОЛЬНИКОВ**  
**Профиль «Информационные технологии»**  
**Командный кейс №6 «Образовательный портал»**

---

- проводите тестирование на каждом этапе разработки;
- регулярно фиксируйте изменения в коде, создавайте ветки для новых функциональностей;
- применяйте ООП для организации кода, для формирования структуры, облегчения сопровождения и повторного использования кода;
- ведите подробную документацию проекта;
- разрабатывайте систему так, чтобы ее можно было легко масштабировать;
- уделите особое внимание вопросам безопасности, включая защиту данных пользователей, шифрование, управление доступом.

#### **4. Требования к документации**

- Титульный лист (с указанием названия кейса и перечислением членов команды).
- Обоснование выбора языка программирования и используемых программных средств.
- Структурная и функциональная схемы программного продукта.
- Блок-схема работы основного алгоритма.
- Описание особенностей и аргументация выбранного типа СУБД.
- Схема базы данных.
- Программный код (ссылка на репозиторий), файл README должен включать:
  - краткое описание проекта;
  - инструкцию по установке/развертыванию;
  - ссылку на видеоролик.

#### **5. Требования к видеоролику**

- Видеоролик должен демонстрировать функционирование разработанного программного продукта в соответствии с регламентом испытаний.
- На видео или записи экрана необходимо продемонстрировать выполнение каждого испытания, описанного в регламенте, в соответствии с условиями.

**МОСКОВСКАЯ ПРЕДПРОФЕССИОНАЛЬНАЯ  
ОЛИМПИАДА ШКОЛЬНИКОВ**  
**Профиль «Информационные технологии»**  
**Командный кейс №6 «Образовательный портал»**

---

- Видео должно однозначно подтверждать авторство участников (во время записи ролика необходимо четко произнести название команды, ФИО участников, номер школы, ФИО руководителя).
- Видеоролик необходимо разместить на стороннем видеохостинге («ВКонтакте», Rutube и др.)

## **6. Регламент испытаний**

1. **Базовое функциональное тестирование:** Проверка основных сценариев работы приложения, таких как регистрация и авторизация пользователей, доступ к курсам и материалам, выполнение простых заданий. Цель — убедиться в корректной работе основных функций без значительных нагрузок.
2. **Интеграционное тестирование:** Тестирование взаимодействия между различными модулями системы, такими как: управление курсами, аналитика успеваемости и форум. Проверка корректности передачи данных и работы зависимостей между модулями.
3. **Нагрузочное тестирование:** Оценка работы системы при увеличении количества одновременных пользователей, выполнении сложных заданий и обработке большого объема данных. Определение пределов производительности и выявление узких мест.
4. **Тестирование критических сценариев:** Проведение тестов на устойчивость системы к критическим ситуациям, таким как потеря соединения с базой данных, сбои в работе серверов, некорректные действия пользователей. Оценка способности системы к восстановлению и минимизации потерь данных.

## **7. Методические материалы**

- **Фронтенд-фреймворк:**
  1. Vue.js: <https://vuejs.org/>
  2. React: <https://reactjs.org/>
- **Бэкенд-платформа:**
  1. Node.js: <https://nodejs.org/>
  2. Express.js: <https://expressjs.com/>
- **База данных:**

**МОСКОВСКАЯ ПРЕДПРОФЕССИОНАЛЬНАЯ  
ОЛИМПИАДА ШКОЛЬНИКОВ**  
**Профиль «Информационные технологии»**  
**Командный кейс №7 «Обслуживание банкоматов»**

---

## 1. Условия

Перед участниками будет стоять задача по разработке веб-приложения для обслуживания сети банкоматов. Команде будет дан блок данных, полученных с нескольких банкоматов, в виде CSV-файла. На предложенном объеме данных команде необходимо будет решить аналитическую задачу по предсказанию вызова службы инкассации для замены кассет с купюрами или инженера для ремонта банкомата, а также построение оптимального пути на карте для инженера или машины службы инкассации.

## 2. Техническое задание

*Функциональное задание:*

Обязательная функциональность:

- Чтение [CSV-файла](#), перенос информации в базу данных в соответствии с временем получения (имитация последовательного получения данных)
- Демонстрация отчетов по полученным данным для отдельного банкомата:
  - процент времени работы банкомата за временные периоды (неделя, месяц);
  - общий список банкоматов с их статусами;
  - график произошедших сбоев за неделю и месяц.
- Анализ предложенного массива данных с целью прогноза необходимости вызова механика для ремонта банкомата, а также вызова инкассации с учетом времени подъезда механиков и инкассации до банкомата.
- Построение кратчайшего маршрута для обслуживания нескольких банкоматов для одного механика или машины инкассации.

Требования к пользовательскому интерфейсу:

- Приложение должна содержать следующие пункты меню:
  - Администрирование
    - Добавить/удалить банкомат
    - Изменить карту
    - Добавить/удалить механика
    - Добавить/удалить машину инкассации
  - Отчетность

**МОСКОВСКАЯ ПРЕДПРОФЕССИОНАЛЬНАЯ  
ОЛИМПИАДА ШКОЛЬНИКОВ**  
**Профиль «Информационные технологии»**  
**Командный кейс №7 «Обслуживание банкоматов»**

---

- Показать отчет о состоянии парка банкоматов (с фильтрами по всем колонкам и постраничным выводом)
- Показать общий список банкоматов с их статусами
- Показать список полученных сообщений от банкоматов
- Аналитические задачи
  - Решение задачи предсказания сбоя в работе банкоматов или необходимости вызова службы инкассации.
  - Решение задачи для механика по обслуживанию нескольких банкоматов
  - Решение задачи для машины службы инкассации для обслуживания нескольких банкоматов

### **3. Рекомендации к выполнению**

- Приложение должно иметь клиент-серверную архитектуру с веб-интерфейсом.
- Выбор БД - на усмотрение команды
- Выбор инструментов для реализации веб-интерфейса - на усмотрение команды
- Выбор языка реализации - на усмотрение команды
- Рекомендуется использовать api картографического сервиса для построения маршрута между банкоматами.
- Разработка должна вестись с применением GIT-репозитория GitVerse.ru (при этом каждый член команды должен сделать не менее трех коммитов для того, чтобы было видно его вклад в общий репозиторий)

### **4. Требования к документации**

- Титульный лист (с указанием названия кейса и перечислением членов команды).
- Обоснование выбора языка программирования и используемых программных средств.
- Структурная и функциональная схемы программного продукта.
- Блок-схема работы основного алгоритма.
- Описание особенностей и аргументация выбранного типа СУБД.
- Схема базы данных.

**МОСКОВСКАЯ ПРЕДПРОФЕССИОНАЛЬНАЯ  
ОЛИМПИАДА ШКОЛЬНИКОВ**  
**Профиль «Информационные технологии»**  
**Командный кейс №7 «Обслуживание банкоматов»**

---

- Программный код (ссылка на репозиторий), файл README должен включать:
  - краткое описание проекта;
  - инструкцию по установке/развертыванию;
  - ссылку на видеоролик.

## **5. Требования к видеоролику**

- Видеоролик должен демонстрировать функционирование разработанного программного продукта в соответствии с регламентом испытаний.
- На видео или записи экрана необходимо продемонстрировать выполнение каждого испытания, описанного в регламенте, в соответствии с условиями.
- Видео должно однозначно подтверждать авторство участников (во время записи ролика необходимо четко произнести название команды, ФИО участников, номер школы, ФИО руководителя).
- Видеоролик необходимо разместить на стороннем видеохостинге («ВКонтакте», Rutube и др.)

**МОСКОВСКАЯ ПРЕДПРОФЕССИОНАЛЬНАЯ  
ОЛИМПИАДА ШКОЛЬНИКОВ**  
**Профиль «Информационные технологии»**  
**Командный кейс №7 «Обслуживание банкоматов»**

---

**6. Регламент испытаний**

1. Запуск приложения. Заполнение необходимых первичных данных.
2. Запуск модуля/утилиты/сервиса импорта данных для имитации последовательного получения данных от парка банкоматов в реальном времени.
3. Запуск отчетов. Каждый отчет запускается по отдельности с демонстрацией всех возможностей модуля отчетности (фильтры, постраничный вывод и т.п.)
4. Запуск аналитических задач
  - a. Отображение предсказания необходимости технического обслуживания банкоматов или вызова для банкомата службы инкассации
  - b. Отображение решения задачи для получения путевого задания для инженера на обслуживание нескольких банкоматов с нахождением кратчайшего пути
  - c. Отображение решения задачи нахождения кратчайшего пути для обслуживания нескольких банкоматов службой инкассации.
  - d. Решение каждого вышеописанного пункта приводится два раза для демонстрации динамического изменения результата из-за поступления новых данных

**7. Методические материалы**

- <http://gitverse.ru>
- <https://www.python.org/>
- <https://sqlite.org/>
- <https://www.postgresql.org/>
- <https://flask.palletsprojects.com/en/2.0.x/>

**МОСКОВСКАЯ ПРЕДПРОФЕССИОНАЛЬНАЯ  
ОЛИМПИАДА ШКОЛЬНИКОВ**  
**Профиль «Информационные технологии»**  
**Командный кейс №7 «Обслуживание банкоматов»**

---

- <https://developer.android.com/>
- <https://kotlinlang.org/docs/home.html>
- <https://docs.oracle.com/en/>
- <https://vuejs.org/>