**МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ**

**Учреждение образования**

**«Гомельский Государственный**

**Технический Университет**

**имени П.О. Сухого»**

**Кафедра «Информационные Технологии»**

**Лабораторная работа**

**по дисциплине**

«ВВЕДНИЕ В ОБЛАЧНЫЕ ВЫЧИСЛЕНИ»

для специальности

"информационные системы и технологии"

Гомель 2021

**Цель работы:** научитьсяразрабатывать программное обеспечение с использованием облачных вычислений, используя для разработки, выполнения приложений и хранения данных облачную платформу «*Microsoft Azure*».

**Задание:**

Разработать программное обеспечение с использованием облачных вычислений. Тему предметной области и описание необходимо выбрать из таблицы 1 в соответствии со своим номером варианта.

Таблица 1 – Предметная область и описание программного обеспечения

|  |  |
| --- | --- |
| Вариант | Предметная область и описание ПО |
| 1,  7,  13,  19,  25 | **Университет.**  Необходимо разработать приложение для студентов и преподавателей ВУЗ’a (или иного учреждения образования).  • Студент должен иметь возможность получать данные о собственной успеваемости: получить оценку по первой и второй аттестации за указанный студентом предмет, получить оценку по заданному экзамену, узнать состояние зачёта по соответствующему предмету, указанному студентом.  Поиск оценок студента осуществляется по его фамилии или номеру студенческого билета (реализовать оба варианта, поиск осуществлять по тому, что укажет пользователь). Результаты должны отображаться только по строго указанному предмету (не нужно выводить весь список сразу).  • Преподаватели должны иметь возможность заполнить эти данные для студента по каждому предмету, иметь возможность редактировать и удалять оценки.  В приложении должна быть возможность получить список студентов, у которых пересдача:   1. в заданный пользователем приложения день; 2. по заданному пользователем приложения предмету. |
| 2,  8,  14,  20,  26 | **Онлайн-библиотека**.  Необходимо разработать приложение для возможности чтения книг.  • Если у пользователя есть подписка, открывающая возможность чтения платных книг, предоставлять пользователю доступ к этим книгам. Чтение бесплатных книг должно быть доступно всем авторизированным пользователям, даже у которых нет наличия подписки. Книги выдавать клиенту порционно, по страницам (то есть с сервера возвращать только часть текста всей книги, согласно номеру страницы, которую читает пользователь). Пользователь должен иметь возможность узнать как долго ещё активна его подписка.  • Администратор должен иметь возможность добавлять в библиотеку книги, редактировать их сведенья (названия, описание, автора, дату выхода, жанр и прочее). Администратор должен иметь возможность предоставлять пользователю подписку на ограниченное время, а также иметь возможность досрочно её аннулировать.  • В приложении должна быть реализована возможность поиска книги по названию, а также выборка книг по жанру, году и/или автору. |
| 3,  9,  15,  21,  27 | **Меню ресторана.**  Необходимо разработать меню для возможности заказа блюда в ресторане быстрого питания.  • Посетитель заведения через специализированный интерфейс должен иметь возможность сформировать корзину товаров из разных категорий блюд (например: бургеры, роллы, салаты, десерты, напитки, горячее и т.д.). У посетителя должна быть возможность получить информацию о статусе готовности его заказа: «готовится» или «готов».  • У диспетчера заведения должна иметься возможность добавлять товары в меню, удалять товары из меню, редактировать их описание (в том числе и цену). Диспетчер заведения должен видеть все заказанные товары со списком всех выбранных пользователем продуктов. У диспетчера должна быть возможность переводить статусы заказа из «готовится» в «готов». У диспетчера должна быть возможность получить информацию о сумме заработка за текущий день (согласно сумме стоимости всех заказов со статусом «готов»). |
| 4,  10,  16,  22,  28 | **Курьерские заказы.**  Необходимо разработать приложение для курьеров.  • У курьера должна быть реализована следующая функциональность:  – возможность отметиться, что он забрал определённый товар из определённого пункта выдачи посылок;  – возможность отметить, что товар доставлен по адресу.  • У диспетчера должна быть возможность:  – указывать пункты выдачи посылок;  – добавлять, удалять и редактировать сведенья о курьерах;  – добавлять, удалять и редактировать информацию о посылках.  • У получателя посылки должна быть возможность узнать состояние своей посылки («ожидает ли курьера в пункте получения посылок», «находится у курьера и доставляется в пункт передачи получателю», или «уже вручена получателю»), а также, если посылка в пути, отобразить ФИО и номер телефона этого курьера. |
| 5,  11,  17,  23,  29 | **Продажа билетов.**  Необходимо разработать программу для продажи билетов.  • Администратор должен иметь возможность добавлять записи о фильмах, редактировать их и удалять записи целиком.  • Пользователь должен иметь возможность выбрать фильм из списка доступных в заданный день, получить информацию о количестве свободных мест на данный сеанс, зарезервировать место или несколько мест, произвести псевдооплату (фактически оплачивать билеты не нужно, достаточно эмулировать этот процесс). После оплаты на электронную почту пользователя должно отправиться письмо с информацией о забронированном месте и сеансе. У пользователя не должно быть возможности выбрать то же место, что уже выбрал другой пользователь, но ещё не оплатил его.  • Сеансы, которые уже были показаны в кинотеатре на настоящий момент времени, не должны более отображаться пользователю. |
| 6,  12,  18,  24,  30 | **Онлайн-тесты.**  Необходимо разработать приложение для прохождения онлайн-тестов.  • Пользователь должен иметь возможность выбрать предметную область теста, пройти тест. Вопросы теста по одному отправлять клиенту (то есть сервер не весь список сразу должен выдать, а выдавать порционно, по одному, подавая новый после подачи ответа на предыдущий). Сервер не должен сообщать клиенту о верных или не верных ответах: он должен получить все ответы на все вопросы, только после чего проанализировать их и выдать только результат теста, указав какие вопросы неверны, но не сообщая верные ответы. Проверка и обработка вопросов должна совершаться только на backend!  • Администратор должен иметь возможность составлять тест и добавлять его в подходящую предметную область, редактировать тесты (вопросы, ответы) и удалять их. |

Общие требования к разрабатываемым проектам:

• всю логику приложения и вычислительные операции реализовать на платформе *Node.JS*;

• в качестве хранилища данных использовать *MongoDB*;

• использовать фреймворк *Express* для организации *API* приложения;

• разработать методы для CRUD-операций (создания, чтения, обновления и удаления данных);

• обязательно организовать систему авторизации и распределения ролей;

• *backend* полученного приложения разместить на платформе *Azure*;

• интерфейс для управления приложением и получения данных с облака допускается реализовать в рамках *android*, *iOs*, *web*-приложения или чат-бота какого-либо мессенджера;

• подготовить подробный отчёт о проделанной работе, с указанием документированного листинга, демонстрацией работы приложения и его возможностей.