

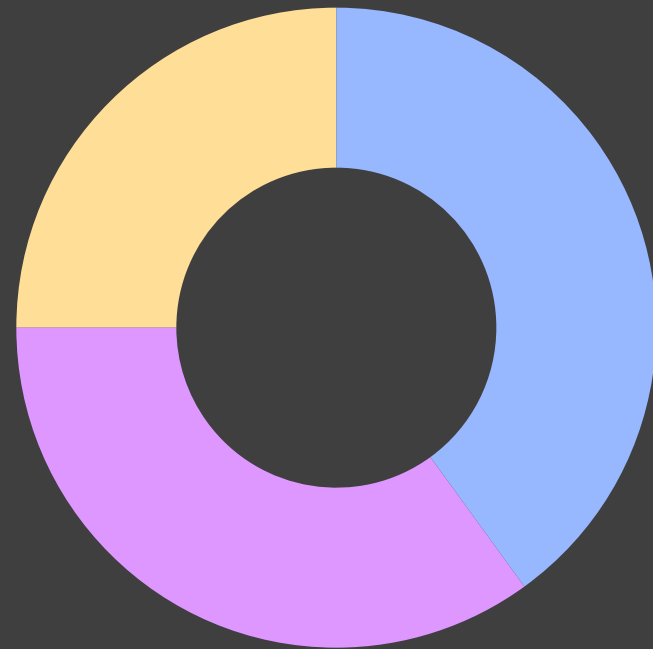
Государственное бюджетное общеобразовательное учреждение города  
Москвы «Школа № 2098 имени Героя Советского Союза Л.М. Доватора»

## «Автоматизированное управление столовой»

Выполнили: Редикальцева Мария - ученица 11 "Т" класса,  
ГБОУ Школа №2098 имени Героя Советского Союза Л.М.  
Доватора, Зборовский Михаил - ученик 11 "Т" класса, ГБОУ  
Школа №2098 имени Героя Советского Союза Л.М.  
Доватора

Научные руководители: Малевин Дмитрий Сергеевич —  
учителя информатики, ГБОУ Школа №2098 имени Героя  
Советского Союза Л.М. Доватора

# Актуальность и цель работы



- Традиционная бумажная система
- Базовая автоматизация (Excel)
- Автоматизированная система с использованием электронных карты

**Цель проекта:** создать автоматизированную веб-систему для школьной столовой, которая заменит бумажный учёт и упростит взаимодействие всех участников: от выбора и оплаты питания учениками до контроля продуктов и получения отчётов.

# Сравнение с аналогами

Система	Преимущества	Недостатки
Наш проект	<ul style="list-style-type: none"><li>• Простота внедрения и гибкие роли</li><li>• Абонементы и учёт запасов</li><li>• Отзывы и безопасные карты</li></ul>	<ul style="list-style-type: none"><li>• Нет интеграции с проходными</li><li>• Ограниченные оплаты и нет приложения</li><li>• Нет разделения счёта и льгот</li></ul>
Москвенок	<ul style="list-style-type: none"><li>• Родители заказывают обеды и контролируют рацион</li><li>• Можно установить лимит и запретить покупки в буфете</li><li>• Карта даёт бесплатный проезд на транспорте</li></ul>	<ul style="list-style-type: none"><li>• Ограничена географией Москвы</li><li>• Настройка лимитов требует портала <a href="https://mos.ru">mos.ru</a></li><li>• Нет отдельных счетов для буфета и обеда</li></ul>
Глолайм	<ul style="list-style-type: none"><li>• Раздельные счета для обеда и буфета</li><li>• Реальное время контроля и удобные пополнения</li><li>• Возможность оплаты транспорта и кружков</li></ul>	<ul style="list-style-type: none"><li>• Региональная доступность и зависимость от инфраструктуры</li><li>• Нет интеграции с электронными дневниками</li><li>• Стоимость оборудования и обслуживания высокая</li></ul>

# Пример взаимодействия пользователя

## Регистрация/вход

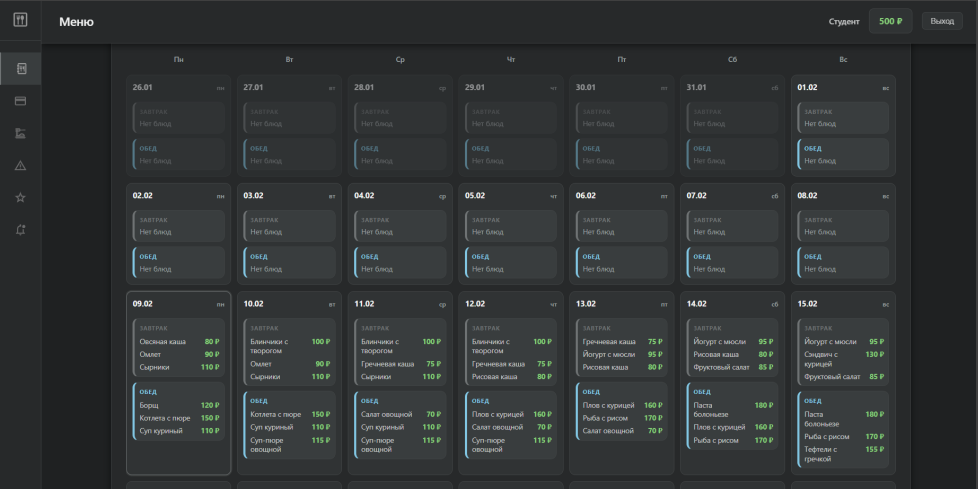
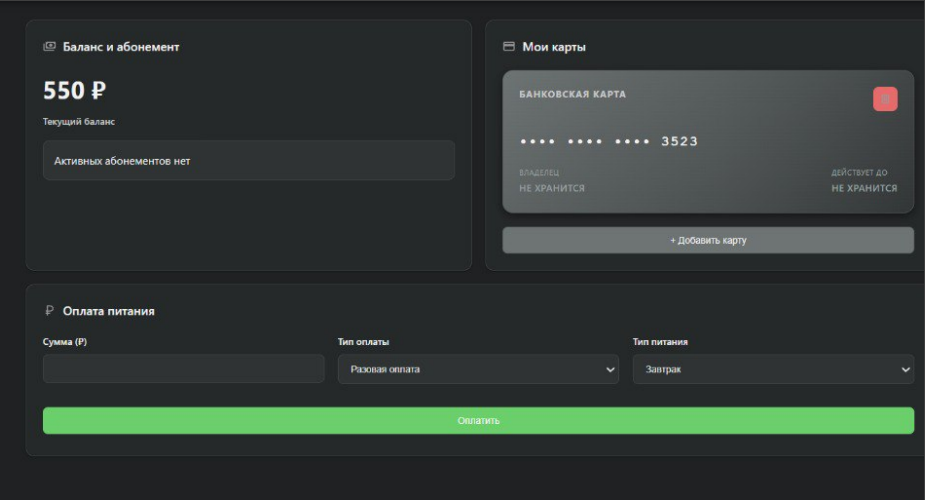
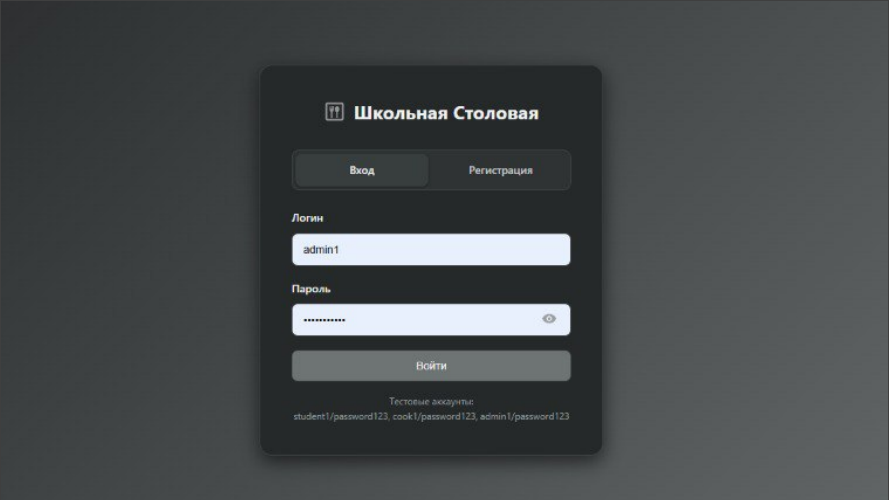
Пользователь регистрируется в аккаунт в зависимости от своей роли

## Оплата питания

Пользователь добавляет свою карту, после чего с нее списывают деньги за то или иное питание

## Календарь с меню

Удобный календарь, с возможностью посмотреть меню на другие дни



# Задачи работы

1 Разработка функционала для трёх групп пользователей

3 Автоматизация ключевых процессов

2 Обеспечение безопасности системы

4 Подготовка к внедрению: проведение тестирования

```
3334 @app.route('/api/meal-claims/today', methods=['GET'])
3335 @login_required
3336 @role_required('student')
3337 def get_today_meal_claims():
3338     """История выдачи питания за сегодня для текущего ученика (включая подтве
3339     student_id = session.get('user_id')
3340
3341     db = get_db()
3342     cursor = db.cursor()
3343
3344     today = datetime.now().date().strftime('%Y-%m-%d')
3345
3346     try:
3347         try:
3348             ensure_column(cursor, 'meal_claims', 'student_received', 'INTEGER')
3349             ensure_column(cursor, 'meal_claims', 'student_marked_at', 'TIMEST
3350         except Exception:
3351             pass
3352
3353         rows = cursor.execute(
3354             '''
3355             SELECT
3356                 mc.id,
3357                 mc.meal_type,
3358                 mc.claimed_at,
3359                 mc.issued_by,
3360                 iu.full_name AS issuer_name,
3361                 mc.menu_item_id,
3362                 mi.name AS dish_name,
3363                 mc.student_received,
3364                 mc.student_marked_at
3365             FROM meal_claims mc
3366             LEFT JOIN users iu ON iu.id = mc.issued_by
3367             LEFT JOIN menu_items mi ON mi.id = mc.menu_item_id
3368             WHERE mc.user_id = ?
3369                 AND DATE(mc.claimed_at) = ?
3370             ORDER BY datetime(mc.claimed_at) DESC, mc.id DESC
3371             ''',
3372                 (student_id, today)
3373             ).fetchall()
3374
3375         return jsonify([dict(r) for r in rows])
3376
3377     except Exception as e:
3378         return jsonify({'error': str(e)}), 500
3379     finally:
3380         db.close()
3381
```

## Сделано на данный момент

1

Разработано полнофункциональное веб-приложение на Python с Flask, включающее все необходимые модули для управления питанием, учётом и аналитикой.

2

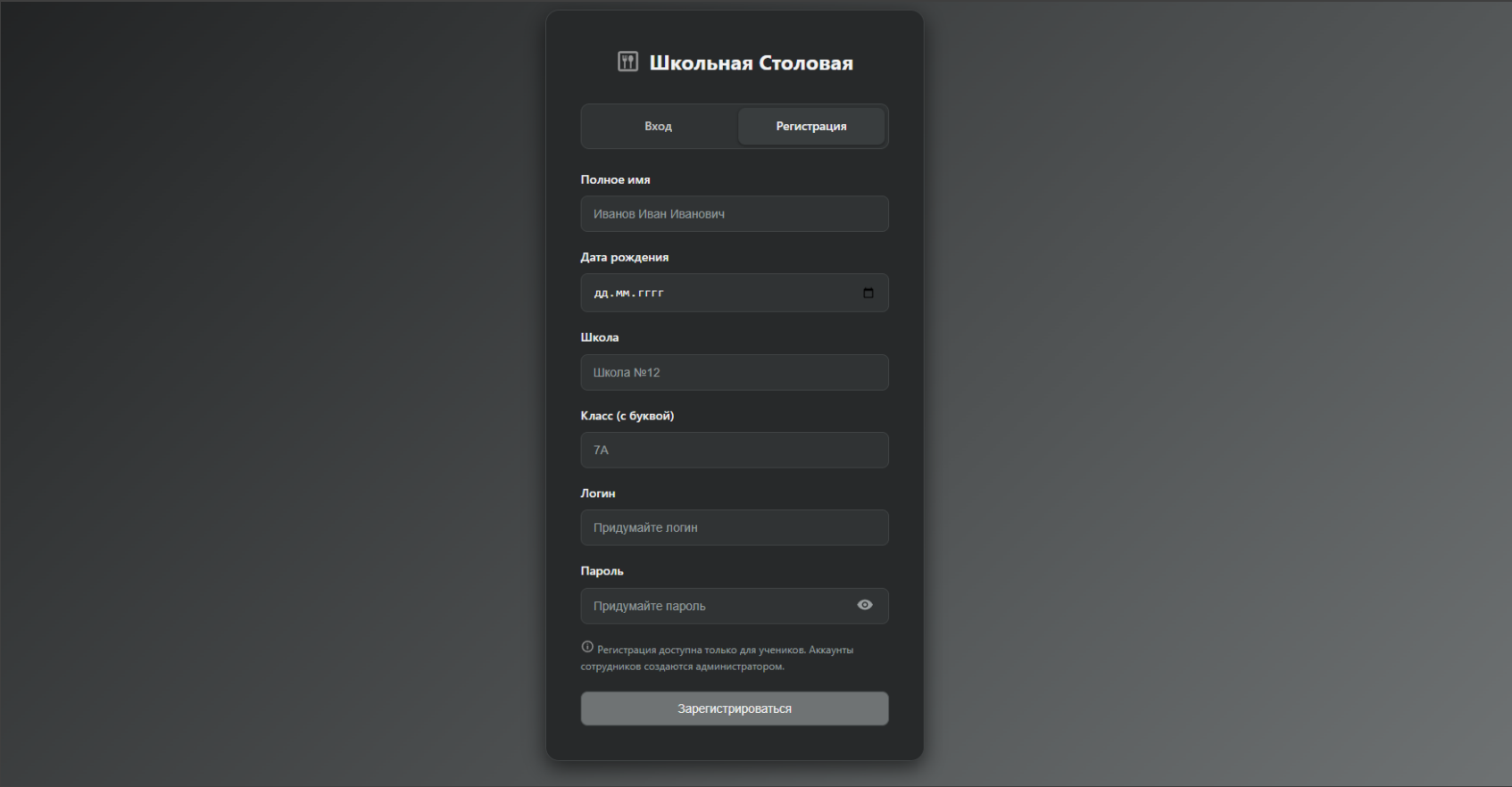
Реализованы персонализированные интерфейсы для трёх ролей пользователей: учеников (выбор и оплата), поваров (учёт выдачи) и администраторов (отчёты и контроль)

3

Обеспечена безопасность и готовность к внедрению благодаря защищённому хранению данных, токенизации платежей и контейнеризации через Docker.

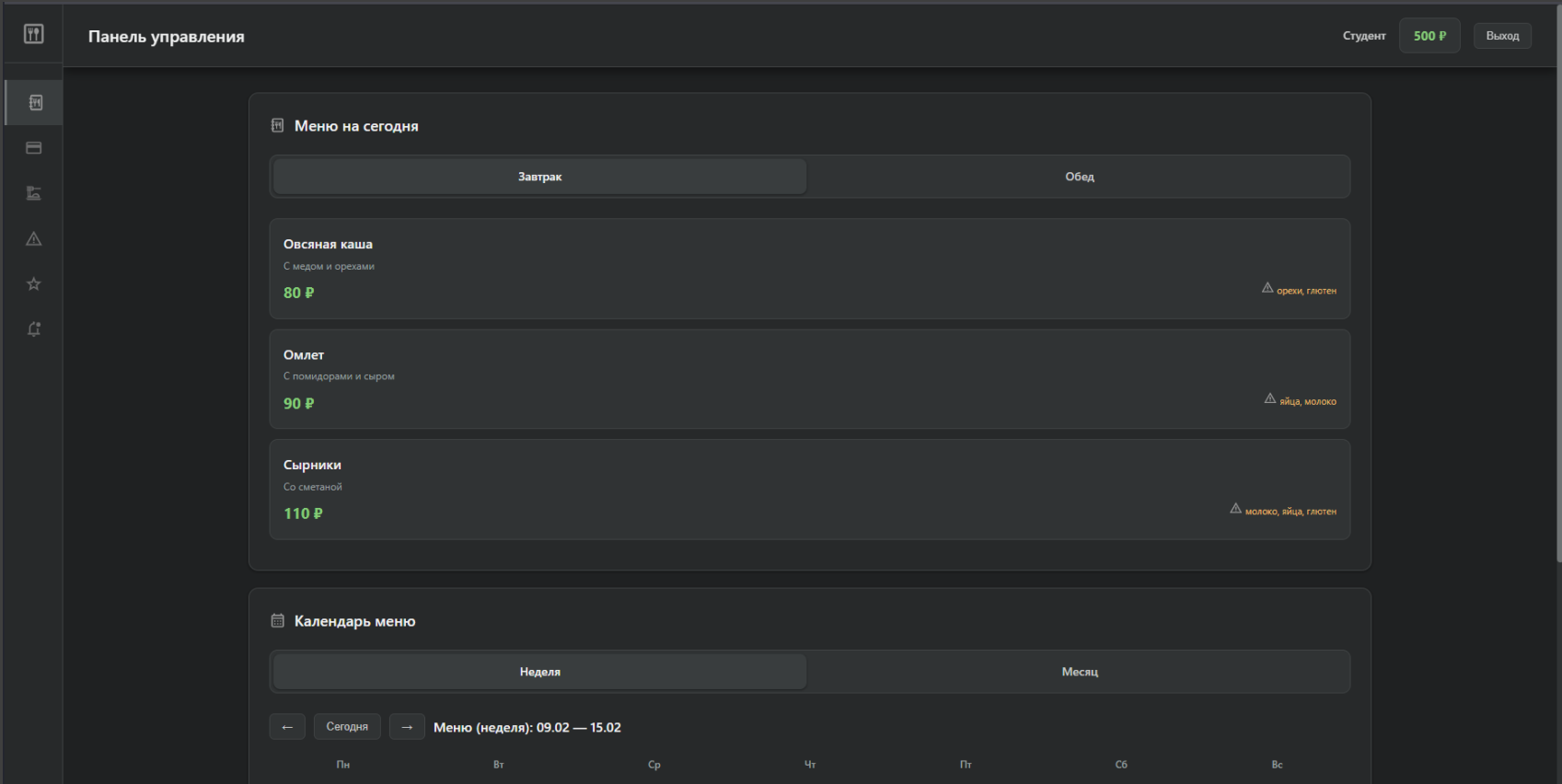
# Страница регистрации(Ученик)

Эта страница доступна только ученикам, она содержит форму для регистрации.



# Главная страница/меню(Ученик)

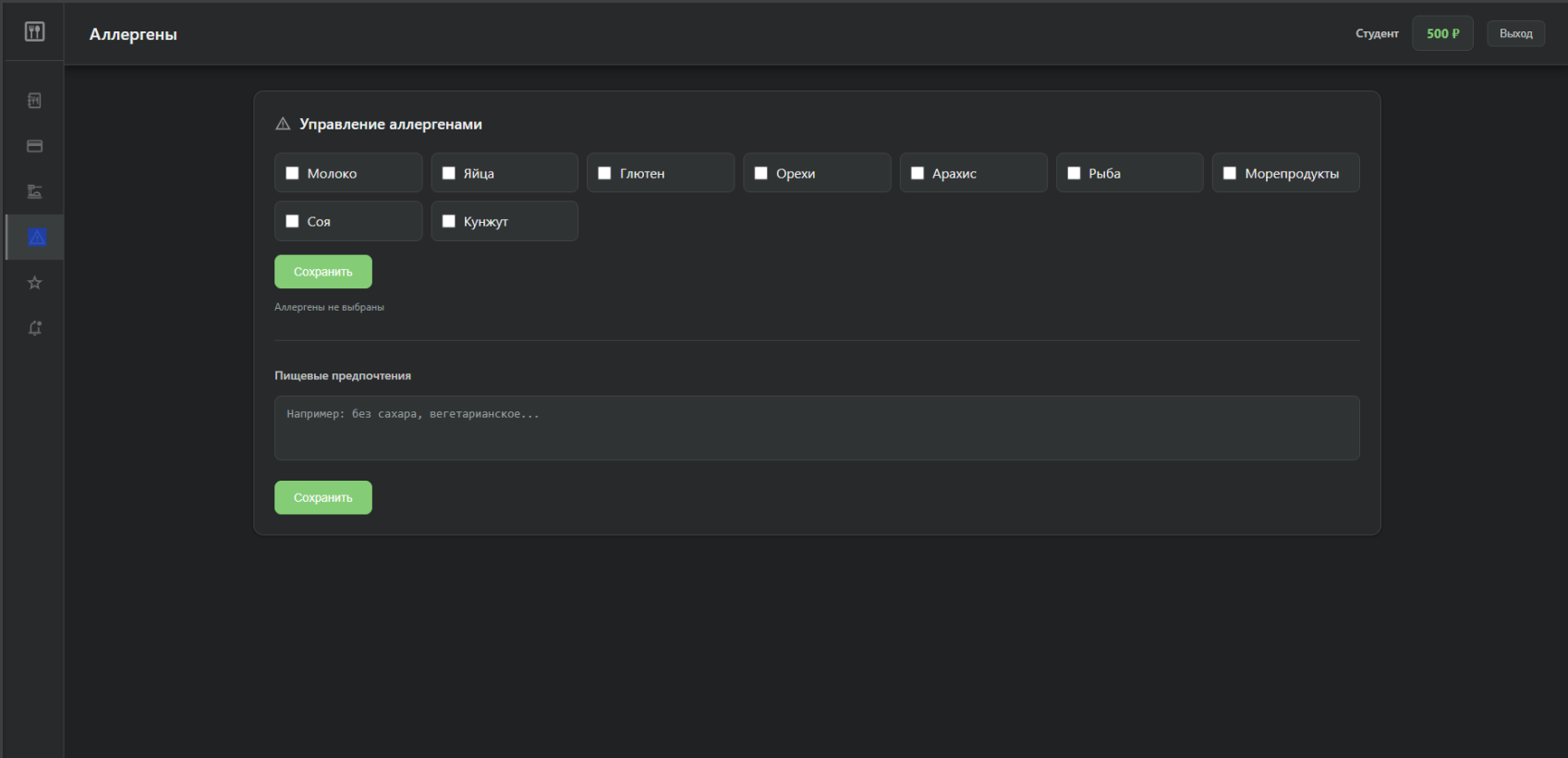
Данная страница содержит  
информацию про меню на сегодняшний  
день, а так же расписание на месяц.





# Страница с аллергиями(Ученик)

На данной странице ученик может добавить свои аллергии, чтобы система могла предупредить о наличии аллергена в блюде



# Страница выдачи(Повар)

Здесь повар отмечает  
выдачу питания ученикам

Выдача

Повар

Выход

Выдача питания

Ученик (ФИО)

Тип питания

Блюдо (на сегодня)

Начните вводить ФИО...

Завтрак

Овсяная каша — 80 Р

Список берётся из меню на сегодня.

Выдать

# История выдачи(Повар)

Повар может отслеживать  
ранее выданные заказы

История

Повар

Выход

История выдачи

Период

7 дней

Тип питания

Все

Записи

Все выдачи

Обновить

Записей: 0. Завтраков: 0, обедов: 0. Подтверждение: ожидает 0, получил 0, не получил 0.

Дата/время	Ученик	Тип	Блюдо	Выдал	Подтверждение
Нет записей за выбранный период					

# Страница продуктов(Повар)

Содержит информацию о наличии  
тех или иных продуктов, показывает  
необходимость закупки

Продукты

Повар

Выход

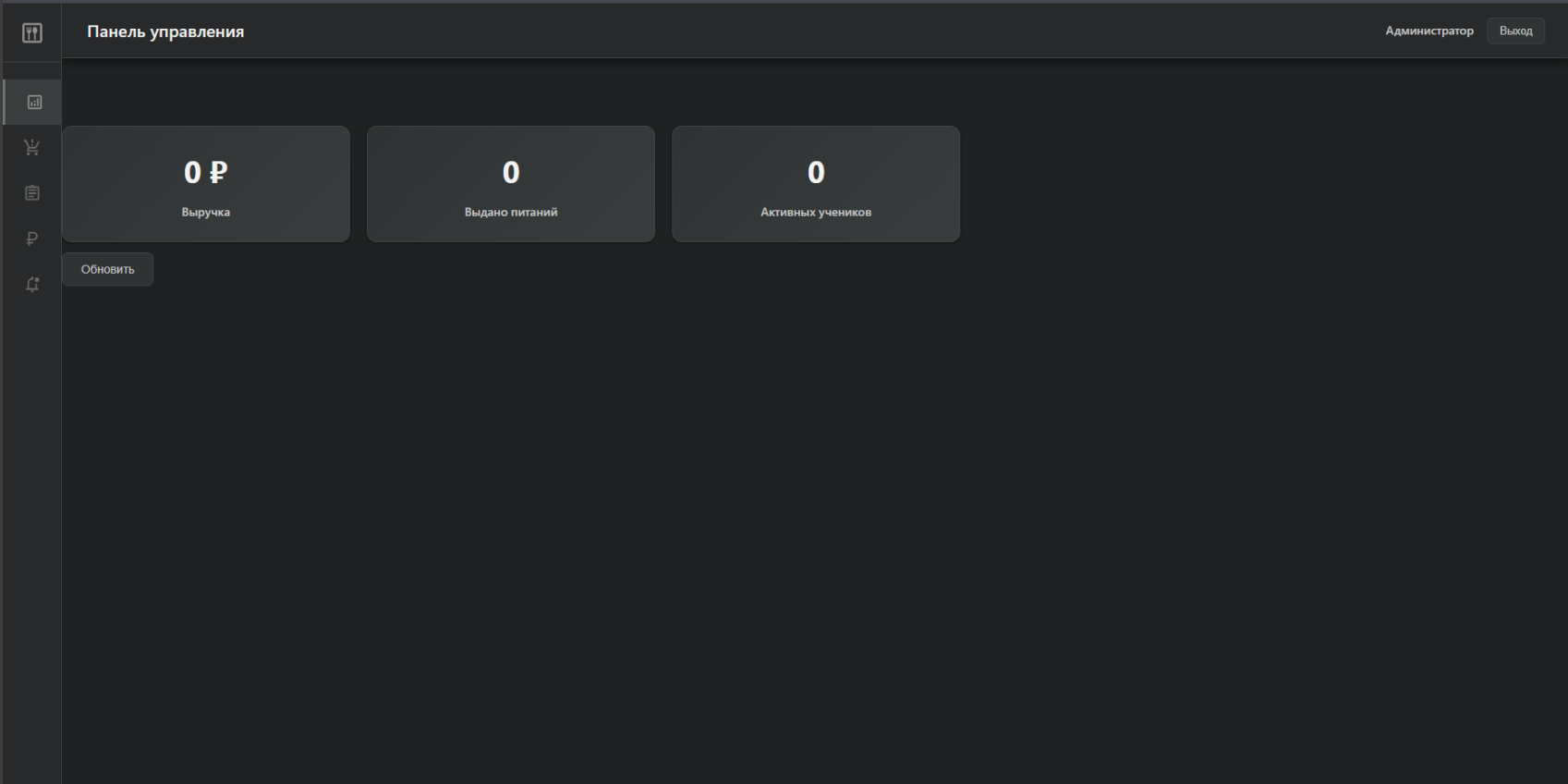
Продукты

Обновить

Продукт	Количество	Мин. остаток	Статус
Мука	50 кг	20 кг	В норме
Молоко	30 л	15 л	В норме
Яйца	100 шт	50 шт	В норме
Овсяные хлопья	25 кг	5 кг	В норме
Мёд	10 кг	2 кг	В норме
Орехи	15 кг	3 кг	В норме
Сахар	50 кг	15 кг	В норме
Соль	20 кг	5 кг	В норме
Сливочное масло	15 кг	5 кг	В норме
Творог	25 кг	8 кг	В норме
Сгущённое молоко	20 л	5 л	В норме
Йогурт натуральный	30 л	10 л	В норме

# Панель управления(Администратор)

Содержит полный отчет о  
выручке, выданных заказов



# Страница заявок(Администратор)

# Заявки о закупке

необходимых продуктов от

# поваров

# Страница уведомлений(Администратор)

Страница с созданием уведомлений для остальных пользователей

Уведомления

Администратор

Выход

🔔 Создать уведомление

Заголовок

Аудитория

Всем

Текст

Отправить

🔔 Уведомления

Обновить

Уведомлений пока нет.

## Перспективы развития

- 1 Расширение использования технологии.
- 2 Внедрение эквайринга.
- 3 Внедрение машинного обучения.





## Школьная Столовая

Вход

Регистрация

Логин

Введите логин

Пароль

Введите пароль



Войти

Тестовые аккаунты:

student1/password123, cook1/password123,  
admin1/password123



192.168.1.128



## Заключение

Разработано веб-приложение «Управление школьной столовой» — целостная система для автоматизации питания в школе, заменяющая бумажный учёт. Приложение работает в браузере, предоставляя интерфейсы для трёх ролей: ученик, повар, администратор.

СПАСИБО ЗА ВНИМАНИЕ!