

Название проекта: Веб-приложение для онлайн-продажи
товаров

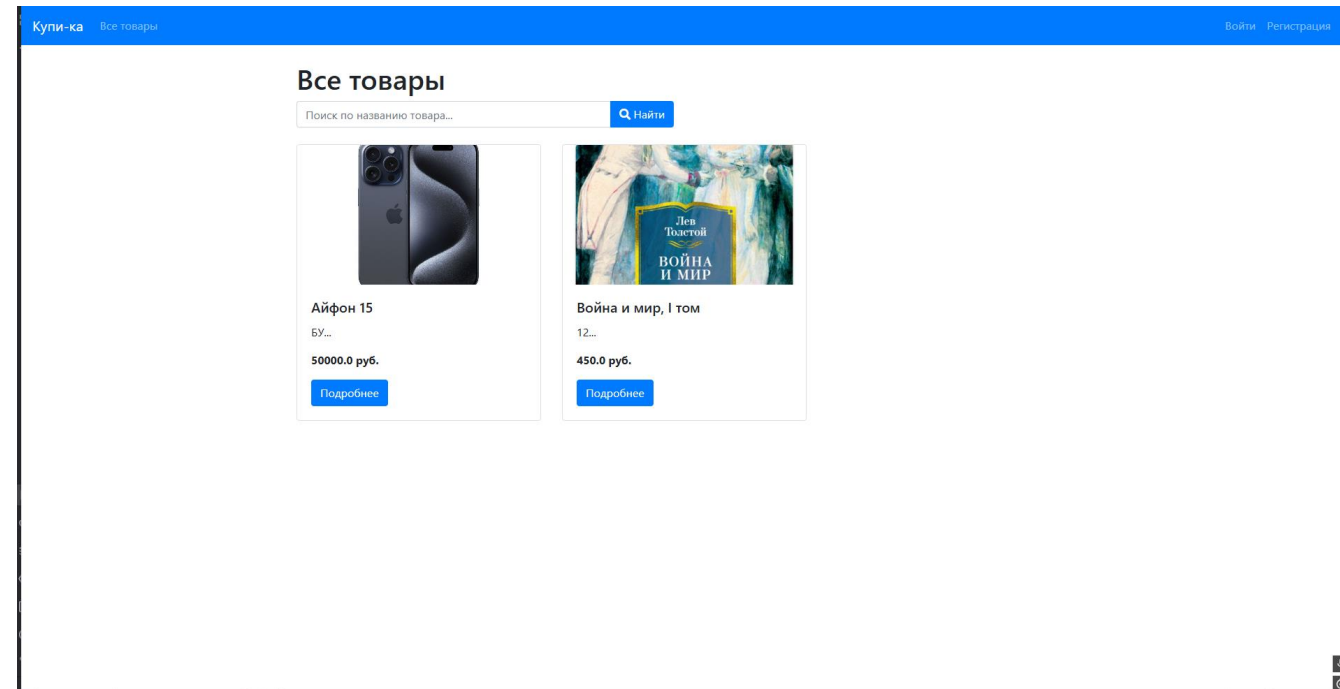
Работу выполнили: Демидов Артем, Лебедев Андрей, Якубов Игорь.

Идея проекта:

Создание удобной платформы для продажи товаров между пользователями.

- Основные задачи:

- ✓ Регистрация и аутентификация пользователей
- ✓ Добавление, редактирование и удаление товаров
- ✓ Оставление отзывов и оценок
- ✓ Совершение покупок
- ✓ Защита данных



Основные модели данных:

- **User** – информация о пользователях
- **Product** – данные о товарах
- **Review** – отзывы и оценки
- **Purchase** – история покупок
- **Связи между моделями:**
- Пользователь ↔ Товары (один ко многим)
- Товар ↔ Отзывы (один ко многим)
- Пользователь ↔ Покупки (один ко многим)

```
import sys

app = Flask(__name__)
app.config.from_object(Config)

db = SQLAlchemy(app)
login = LoginManager(app)
login.login_view = 'login'

class User(UserMixin, db.Model):
    id = db.Column(db.Integer, primary_key=True)
    username = db.Column(db.String(64), index=True, unique=True)
    email = db.Column(db.String(120), index=True, unique=True)
    password_hash = db.Column(db.String(128))
    products = db.relationship('Product', backref='author', lazy='dynamic')
    reviews = db.relationship('Review', backref='author', lazy='dynamic')
    purchases = db.relationship('Purchase', backref='buyer', lazy='dynamic')

    def set_password(self, password):
        self.password_hash = generate_password_hash(password)

    def check_password(self, password):
        return check_password_hash(self.password_hash, password)

class Product(db.Model):
    id = db.Column(db.Integer, primary_key=True)
    title = db.Column(db.String(140))
    description = db.Column(db.Text)
    price = db.Column(db.Float)
    image = db.Column(db.String(200))
    timestamp = db.Column(db.DateTime, index=True, default=datetime.utcnow)
    user_id = db.Column(db.Integer, db.ForeignKey('user_id'))
    category = db.Column(db.String(50))
    reviews = db.relationship('Review', backref='product', lazy='dynamic')
    purchases = db.relationship('Purchase', backref='product', lazy='dynamic')
```

Основные технологии:

- **Flask** – микрофреймворк для обработки запросов
- **Flask-SQLAlchemy** – работа с базой данных
- **Flask-Login** – управление аутентификацией
- **Flask-WTF** – валидация форм
- **Pillow** – обработка изображений
- **Дополнительно:**
- Werkzeug (хеширование паролей)
- Dotenv (загрузка конфигурации)

```
class LoginForm(FlaskForm): 1 usage
    email = StringField(label='Email', validators=[DataRequired(), Email()])
    password = PasswordField(label='Пароль', validators=[DataRequired()])
    remember_me = BooleanField('Запомнить меня')
    submit = SubmitField('Войти')

class RegistrationForm(FlaskForm): 1 usage
    username = StringField(label='Имя пользователя', validators=[DataRequired()])
    email = StringField(label='Email', validators=[DataRequired(), Email()])
    password = PasswordField(label='Пароль', validators=[DataRequired()])
    password2 = PasswordField(
        label='Повторите пароль', validators=[DataRequired(), EqualTo('password')])
    submit = SubmitField('Зарегистрироваться')

    def validate_username(self, username):
        user = User.query.filter_by(username=username.data).first()
        if user is not None:
            raise ValidationError('Пожалуйста, используйте другое имя пользователя.')

    def validate_email(self, email):
        user = User.query.filter_by(email=email.data).first()
        if user is not None:
            raise ValidationError('Пожалуйста, используйте другой email.')

class ProductForm(FlaskForm): 2 usages
    title = StringField(label='Название', validators=[DataRequired(), Length(max=140)])
    description = TextAreaField(label='Описание', validators=[DataRequired()])
    price = FloatField(label='Цена (руб)', validators=[DataRequired()])
    image = FileField(label='Фото товара', validators=[
        FileAllowed(upload_set=['jpg', 'jpeg', 'png', 'gif', 'bmp', 'webp', 'tiff', 'svg'],
            message='Разрешены только изображения!')
    ])
    category = SelectField(label='Категория', choices=[
```

Технологии (Frontend и БД)

- **Frontend:**
- HTML + Jinja2 (шаблонизация)
- Простые формы без JavaScript (для MVP)
- **База данных:**
- SQLite (по умолчанию)
- Поддержка других СУБД через DATABASE_URL

Особенности реализации:

- **Безопасность:**
- Хеширование паролей
- Проверка прав доступа (например, только автор может редактировать товар)
- **Работа с изображениями:**
- Сжатие и конвертация в JPEG
- Автоматическая коррекция ориентации
- **Дополнительные функции:**
- Поиск товаров
- Пагинация

Заключение:

- **Итоги:**
 - ✓ Реализована базовая функциональность площадки для купли-продажи товаров.
 - ✓ Обеспечена безопасность данных.
 - ✓ Простое и удобное управление товарами.
- **Перспективы развития:**
 - Добавление чата с продавцов
 - Интеграция платежных систем
 - Улучшение интерфейса (React/Vue.js)