

Федеральное государственное образовательное бюджетное  
учреждение высшего образования  
«Финансовый университет при Правительстве Российской Федерации»  
(Финансовый университет)

Красноярский филиал Финуниверситета

УТВЕРЖДАЮ

Заместитель директора  
по учебно – методической работе  
Красноярского филиала  
Финуниверситета

В.Г. О.С. Вергейчик  
« 02 » апреля 2026 г.

**ФОНД ОЦЕНОЧНЫХ СРЕДСТВ**

по профессиональному модулю

**ПМ. 01 ПРОЕКТИРОВАНИЕ И РАЗРАБОТКА ИНФОРМАЦИОННЫХ  
РЕСУРСОВ**

---

(код, наименование)

09.02.09 Веб-разработка

---

(код, наименование специальности)

г. Красноярск – 2026

Фонд оценочных средств по профессиональному модулю ПМ. 01 Проектирование и разработка информационных ресурсов разработан на основании федерального государственного образовательного стандарта среднего профессионального образования по специальности 09.02.09 Веб-разработка

Составители:

Цирулькевич Алёна Викторовна. преподаватель

Фонд оценочных средств по дисциплине рассмотрен и рекомендован к утверждению на заседании предметной (цикловой) комиссии общепрофессиональных дисциплин.

Протокол от «02» сентября 2026 г. № 8

Председатель предметной (цикловой)  
комиссии

  
(подпись)

О.А. Полтавец  
(инициалы, фамилия)

# 1. Паспорт фонда оценочных средств по профессиональному модулю ПМ. 01 Проектирование и разработка информационных ресурсов

| Результаты обучения<br>(знания, умения)   | Общие и профессиональные компетенции | Наименование элементов профессионального модуля, раздела, темы  | Наименование оценочного средства   |   |
|---|--------------------------------------|---|--|---|
|   |                                      |   | Текущий контроль   | Промежуточная аттестация                          |
| <p>Знать:</p> <ul style="list-style-type: none"> <li>- основы теории системного анализа и построения концептуальных моделей средствами графических нотаций;</li> <li>- понятия, классификацию информационных систем и ресурсов;</li> <li>- этапы, принципы и особенности проектирования ИС и ресурсов;</li> <li>- архитектуру ИС и ресурсов;</li> <li>- модели процесса разработки ИС и ресурсов.</li> </ul> <p>Уметь:</p> <ul style="list-style-type: none"> <li>- применять методы системного анализа;</li> <li>- интерпретировать бизнес-требования заказчика;</li> <li>- разрабатывать концептуальную модель средствами графических нотаций.</li> </ul> | ОК 01 – 09<br>ПК 1.1                 | <p>МДК 01.01<br/>Проектирование информационных ресурсов</p> <p>- Тема 1.1<br/>Проектирование информационных ресурсов</p>          | <ul style="list-style-type: none"> <li>- Защита практических работ №1, №2.</li> <li>- Выполнение и защита индивидуальных заданий (СР).</li> <li>- Экспертное наблюдение за выполнением заданий.</li> </ul> | Дифференцированный зачет по МДК 01.01             |
| <p>Знать:</p> <ul style="list-style-type: none"> <li>- принципы проектирования пользовательских интерфейсов;</li> <li>- элементы управления пользовательского интерфейса.</li> </ul> <p>Уметь:</p> <ul style="list-style-type: none"> <li>- разрабатывать прототипы пользовательских интерфейсов с использованием UI/UX подхода.</li> </ul>   | ОК 01 – 09<br>ПК 1.2                 | <p>МДК 01.02<br/>Разработка интерфейсов пользователя</p> <p>- Тема 2.1<br/>Разработки прототипов пользовательских интерфейсов</p> | <ul style="list-style-type: none"> <li>- Защита практических работ №1-6.</li> <li>- Выполнение и защита индивидуальных заданий (СР).</li> <li>- Экспертное наблюдение за выполнением заданий.</li> </ul>   | Комплексный дифференцированный зачет по МДК 01.02 |

|  |   |  |   |  |
|--|---|--|---|--|
| <p>Знать:</p> <ul style="list-style-type: none"> <li>- архитектуру ИС и ресурсов;</li> <li>- современные методики тестирования;</li> <li>- принцип устройства и интерфейсы систем хранения версий кода.</li> </ul> <p>Уметь:</p> <ul style="list-style-type: none"> <li>- выбирать и комбинировать техники тестирования;</li> <li>- тестировать ресурс с использованием тест-планов;</li> <li>- применять инструменты подготовки тестовых данных;</li> <li>- создавать отчет по результатам тестирования;</li> <li>- создавать, клонировать репозитории;</li> <li>- создавать ветки и управлять изменениями кода;</li> <li>- решать конфликты версий.</li> </ul> | <p>ОК 01 – 09</p> <p>ПК 1.3</p> <p>ПК 1.4</p> <p>ПК 1.5</p> | <p>МДК 01.03</p> <p>Тестирование информационных ресурсов</p> <p>- Тема 3.1</p> <p>Тестирование готового программного кода</p> <p>- Тема 3.2. Работа с системой контроля версий</p> | <p>- Защита практических работ №1-7.</p> <p>- Выполнение и защита индивидуальных заданий (СР).</p> <p>- Экспертное наблюдение за выполнением заданий.</p> | <p>Комплексный дифференцированный зачет по МДК 01.03</p> |
| <p>Иметь практический опыт:</p> <ul style="list-style-type: none"> <li>- проектирования ИС и ресурсов;</li> <li>- разработки прототипов пользовательских интерфейсов;</li> <li>- разработки тестовых сценариев и тестирования;</li> <li>- документирования результатов тестирования;</li> <li>- работы с системой контроля версий, в т.ч. при коллективной разработке.</li> </ul>  | <p>ОК 01 – 09</p> <p>ПК 1.1 – ПК 1.5</p>                    | <p>Учебная практика</p>  | <p>- Защита отчетов по практике.</p> <p>- Экспертное наблюдение за выполнением работ.</p>   | <p>Дифференцированный зачет</p>                          |

## 2. Формы промежуточной аттестации по профессиональному модулю

| Элементы профессионального модуля                | Формы промежуточной аттестации |
|--|--------------------------------|
| МДК 01.01 Проектирование информационных ресурсов | Дифференцированный зачет       |

|  |                                      |
|--|--------------------------------------|
| МДК 01.02 Разработка интерфейсов пользователя  | Комплексный дифференцированный зачет |
| МДК 01.03 Тестирование информационных ресурсов | Комплексный дифференцированный зачет |
| Учебная практика                               | Дифференцированный зачет             |
| ПМ   | Экзамен по модулю                    |

### 3. Комплект оценочных средств

#### 1. для текущего контроля успеваемости

#### МДК 01.01 Проектирование информационных ресурсов

#### Практическая работа №1

Тема: «Системный анализ деятельности организации и формирование бизнес-требований»

#### 1. Цель работы

Приобрести практические навыки системного анализа предметной области, сбора и интерпретации бизнес-требований заказчика для последующего проектирования информационного ресурса.

#### 2. (Индивидуальный вариант)

Вам необходимо выступить в роли системного аналитика. Выберите одну из предложенных организаций (или получите свой вариант у преподавателя):

- 1) Сеть кофеен «Кофе с собой» (возможность заказа через веб-сайт).
- 2) Ветеринарная клиника «Айболит» (запись на прием и ведение карт пациентов).
- 3) Фитнес-клуб «Здоровая спина» (учет посещений и личный кабинет клиента).
- 4) Агентство недвижимости «Дом» (каталог объектов и заявка на просмотр).

#### 3. Этапы выполнения и содержание отчета

##### Этап 1. Сбор и анализ информации (анализ предметной области)

- Опишите выбранную организацию: сфера деятельности, основные услуги, клиенты, конкуренты (кратко).
- Выделите основные бизнес-процессы, протекающие в организации (не менее 3-х). Используйте текстовое описание (например: «Клиент приходит в кофейню -> Делает заказ -> Бариста готовит -> Клиент платит»).
- Выявите «узкие места» и проблемы текущей реализации процессов (например: «Очереди из-за ручного расчета», «Теряются записи клиентов в бумажной тетради»).

##### Этап 2. Интерпретация бизнес-требований заказчика

Представьте, что владелец бизнеса сформулировал запрос: «Хочу веб-сайт, чтобы ускорить работу и повысить лояльность клиентов».

- Преобразуйте этот общий запрос в список функциональных требований (что именно система должна делать).

а) Пример: «Система должна позволять клиенту просматривать меню онлайн», «Система должна отправлять уведомление бариста о новом заказе».

- Сформулируйте нефункциональные требования (как система должна работать).

а) Пример: «Сайт должен загружаться не более 3 секунд», «Интерфейс должен быть понятен людям старше 50 лет».

### Этап 3. Формирование технического задания (фрагмент)

– На основе собранных требований составьте раздел «Назначение и цели создания системы» и раздел «Требования к функциям (задачам)» для будущего Технического Задания.

### 4. Результат (Отчет)

Оформить текстовый отчет, содержащий:

- 1) Характеристику организации.
- 2) Описание выявленных проблем и «узких мест».
- 3) Перечни функциональных и нефункциональных требований.
- 4) Фрагмент Технического задания.

## Практическая работа №2

Тема: «Построение концептуальной модели информационного ресурса в графических нотациях»

### 1. Цель работы

Освоить практические навыки построения концептуальных моделей информационных ресурсов с использованием графических нотаций (UML, BPMN) на основе требований, сформулированных в работе №1.

### 2. (Сквозное, продолжение ПР №1)

Для той же организации, которую вы анализировали в практической работе №1, разработать комплект графических моделей, описывающих архитектуру и логику работы будущего веб-приложения.

### 3. Этапы выполнения и содержание отчета

Инструментарий: Рекомендуется использовать [draw.io](https://draw.io), MS Visio, ARIS Express или Lucidchart.

#### Этап 1. Моделирование вариантов использования (Use Case Diagram - UML)

- Постройте диаграмму вариантов использования.
- Определите действующих лиц (актеров): Клиент, Администратор, Бариста (или сотрудник).
- Покажите, какие функции системы доступны каждому актеру (на основе функциональных требований из ПР №1).
- Пример: Актер «Клиент» может: «Просмотреть каталог», «Оставить заявку», «Оплатить онлайн».

#### Этап 2. Моделирование бизнес-процесса (BPMN или Activity Diagram - UML)

- Выберите один ключевой процесс (например, «Обработка заказа клиента» или «Запись на прием»).
- Детально опишите его логику в виде диаграммы активности (UML Activity Diagram) или в нотации BPMN.
- На диаграмме должны быть отражены: пулы и дорожки (кто за что отвечает), события (начало/конец), действия, точки принятия решений (ветвления) и потоки данных.

– Постройте модель двух состояний: AS-IS (как процесс выглядит сейчас, без системы) и TO-BE (как процесс будет автоматизирован с помощью вашего веб-ресурса).

Этап 3. Описание архитектуры (Component/Deployment Diagram - UML)

– Предложите архитектуру будущего веб-приложения.  
– Постройте диаграмму компонентов (Component Diagram), показывающую основные части системы: База данных, Серверная часть (Backend), Клиентская часть (Frontend), Платежный шлюз (если есть).  
– Укажите, как эти компоненты взаимодействуют между собой.

4. Результат (Отчет)

1) Файл с графическими схемами (экспортированный в PDF или ссылка на онлайн-проект).  
2) Пояснительная записка к схемам (описание актеров, суть процессов, почему выбрана такая архитектура).

МДК 01.02 Разработка интерфейсов пользователя

Практическая работа №1

Тема: «Знакомство с сервисом Figma. Основы работы»

1. Цель работы:

Изучить интерфейс онлайн-сервиса Figma, освоить базовые инструменты создания и сохранения проектов, навигацию по холсту.

2. Оснащение:

– ПК с доступом в интернет.  
– Браузер (рекомендуется Google Chrome).  
– Учетная запись (аккаунт) в Figma ([figma.com](https://figma.com)).

3.

1) Зарегистрируйтесь в Figma (или войдите в существующий аккаунт).  
2) Создайте новый файл (Draft → New file).  
3) Изучите интерфейс: Панель инструментов (слева), Слои (слева), Свойства (справа), Холст (по центру).

4) С помощью инструментов Frame (Фрейм) создайте монтажную область размером под типовой ноутбук (например, 1440×1024).

5) С помощью инструментов Rectangle, Ellipse, Line нарисуйте простую композицию: «Дом у дороги» (прямоугольник-стена, треугольник-крыша, квадрат-окно, линия-дорога).

6) Экспериментируйте с заливкой (Fill) и обводкой (Stroke) объектов, используя разные цвета.

4. Результат (Отчет):

1) Скриншот созданной композиции в Figma.  
2) Краткое описание в тетради: какие инструменты и панели были использованы.

Практическая работа №2

Тема: «Инструменты. Фрейм. Группа»

### 1. Цель работы:

Научиться создавать фреймы под различные устройства (десктоп, планшет, телефон), работать с группами объектов и выравниванием.

2.

1) В том же файле создайте три фрейма с помощью инструмента Frame (или через раздел Design в свойствах):

– tDesktop (1440×1024)

– Tablet (768×1024)

– Mobile (375×667)

2) Внутри каждого фрейма создайте карточку товара (прямоугольник + круг + текст).

3) Сгруппируйте элементы каждой карточки (Ctrl+G / Cmd+G).

4) Продублируйте карточку несколько раз внутри каждого фрейма.

5) Используя инструменты выравнивания (Align), расположите карточки аккуратно по сетке внутри фреймов.

### 3. Результат (Отчет):

Скриншот экрана Figma, на котором видны три фрейма с аккуратно размещенными карточками товаров.

### Практическая работа №3

Тема: «Слои. Маски. Булевы операции»

### 1. Цель работы:

Освоить работу со слоями, научиться создавать сложные формы с помощью булевых операций (Union, Subtract, Intersect) и применять маски.

2.

1) В новом фрейме создайте иконку «Уведомление» (колокольчик) используя только простые фигуры (круги, прямоугольники) и булевы операции.

2) Создайте иконку «Сердце» (для «лайка») с помощью булевых операций.

3) Найдите в интернете любую картинку (например, фото кота). Вставьте её в Figma (через Shift+Ctrl+K / Copy-Paste).

4) С помощью инструмента Mask (Маска) создайте из этой картинки круглый аватар: поместите картинку в круг, нажмите «Use as mask».

5) Потренируйтесь менять порядок слоев (Forward/Backward), чтобы одни объекты перекрывали другие.

### 3. Результат (Отчет):

Скриншот фрейма, содержащего созданные иконки (колокольчик, сердце) и круглый аватар из фотографии.

### Практическая работа №4

Тема: «Привязки. Компоненты»

### 1. Цель работы:

Изучить механизм работы с компонентами (стилями и многократными элементами) и настроить отзывчивую верстку с помощью Constraints (привязок).

2.

1) Создайте новый фрейм (Desktop).

2) Нарисуйте карточку товара (прямоугольник, картинка-заглушка, название, цена, кнопка «Купить»).



3) Превратите эту карточку в Component (Main Component). Теперь она появится в панели Assets.

4) Создайте несколько экземпляров (Instances) этого компонента на фрейме.

5) В одном из экземпляров измените цвет кнопки и текст — изменения произойдут только в нем (переопределение свойств).

6) Создайте кнопку-компонент с состояниями (Default, Hover, Pressed). Для этого используйте варианты (Variants) в свойствах компонента.

7) Настройте привязки (Constraints) для элементов внутри карточки: например, сделайте так, чтобы картинка всегда была привязана к левому верхнему углу, а кнопка — к правому нижнему. Измените размер фрейма-карточки и посмотрите, как ведут себя элементы.

3. Результат (Отчет):

Скриншот с демонстрацией компонентов и настройками привязок.

Практическая работа №5

Тема: «Адаптация под различные экраны»

1. Цель работы:

Научиться использовать Auto Layout для создания адаптивных макетов, которые корректно отображаются на разных устройствах.

2.

1) Создайте три фрейма: Desktop (1440), Tablet (768), Mobile (375).

2) В десктопном фрейме создайте горизонтальное меню навигации. Используйте Auto Layout для расположения пунктов меню (Главная, Каталог, Контакты).

3) Под меню создайте «Сетку товаров». Для этого:

– Создайте карточку товара (компонент из ПР №4).

– Создайте новый фрейм для контейнера товаров, примените к нему Auto Layout (вертикальный).

– Внутри этого контейнера вложите другой фрейм (горизонтальный Auto Layout) с несколькими карточками.

– Настройте отступы (Padding) и промежутки (Gap between items).

4) Ваша задача: Скопируйте весь блок (меню + контейнер с товарами) в планшетный и мобильный фреймы. Используя настройки Auto Layout и Resizing (Fill container, Hug contents), добейтесь, чтобы на планшете карточки отображались в две колонки, а на телефоне — в одну колонку, занимая всю ширину экрана.

3. Результат (Отчет):

Скриншоты трех фреймов (Desktop, Tablet, Mobile), демонстрирующие корректное адаптивное поведение макета.

Практическая работа №6

Тема: «Разработать сайтмэп и прототипы пользовательских интерфейсов с использованием UI/UX подхода»

(Итоговая, сквозная работа)

1. Цель работы:

Разработать полный проект интерфейса для веб-приложения, применяя все полученные навыки и принципы UI/UX-дизайна.

2. (Индивидуальный вариант):

Выберите тему для разработки (или получите вариант у преподавателя):

- 1) Интернет-магазин одежды/обуви.
- 2) Сайт портфолио фотографа/дизайнера.
- 3) Блог о путешествиях.
- 4) Личный кабинет фитнес-трекера (прогресс тренировок).
- 5) Сайт кинотеатра с расписанием.

3. Этапы выполнения и состав проекта:

Этап 1. Анализ и структура

- Проведите анализ 2-3 сайтов-конкурентов в выбранной теме. Выпишите, какие решения вам нравятся, а какие — нет.
- Определите целевую аудиторию вашего сайта.
- Разработайте сайтмэп (Site Map) — структуру страниц вашего будущего сайта. Выполните в Figma в виде схемы.

Этап 2. Low-fidelity прототипы (Вайрфреймы)

- Создайте черно-белые схемы (вайрфреймы) основных страниц: Главная, Каталог/Список, Карточка товара/поста, Контакты.
- Не используйте цвета и детальные картинки. Ваша задача — показать расположение основных блоков (шапка, меню, контент, подвал).

Этап 3. High-fidelity прототип (Дизайн-концепция)

- Разработайте гайдлайн вашего проекта: выберите цветовую палитру (3-4 цвета), шрифты (заголовочный и текстовый), стили кнопок, карточек, полей ввода. Сохраните их как Styles в Figma (Color Styles, Text Styles).
- Примените стили к вашим вайрфреймам. Добавьте иконки и реалистичные картинки-заглушки.
- Создайте дизайн минимум для 3 страниц в «чистовом» варианте (Desktop версия).

Этап 4. Адаптивность и интерактивность

- Сделайте адаптивные версии (Tablet, Mobile) для главной страницы.
- Используя компоненты и Auto Layout, убедитесь, что макет «дышит».
- Добавьте интерактивность (Prototype). Свяжите кнопки так, чтобы можно было «кликнуть» по ссылке в меню и перейти на другую страницу, нажать кнопку «Купить» и увидеть всплывающее окно (pop-up).

Этап 5. Демонстрация и защита

- Подготовьте презентацию своего проекта. Расскажите, для кого этот сайт, какие задачи он решает, почему вы выбрали такие цвета и расположение блоков.
- Продемонстрируйте работу прототипа в режиме презентации (Play).

4. Результат (Отчет):

- 1) Ссылка на проект Figma с предоставленным доступом на просмотр.
- 2) Краткая пояснительная записка (в электронном виде), описывающая ЦА, гайдлайн и основные UX-решения.

## Практическая работа №1

Тема: «Функциональное тестирование»

### 1. Цель работы:

Освоить методику функционального тестирования, научиться проверять соответствие программы заявленным требованиям и составлять тест-кейсы.

### 2. Оснащение:

- ПК с доступом в интернет.
- Простой веб-калькулятор (например, <https://www.calculator.net/> или любой другой) или калькулятор в ОС Windows.
- Текстовый редактор для оформления отчета.

### 3. :

1) Изучите функционал калькулятора (сложение, вычитание, умножение, деление, работа с отрицательными числами, деление на ноль).

2) Составьте 10 тест-кейсов (минимум), покрывающих:

- Позитивные сценарии ( $2+2=4$ ,  $5-3=2$ ).
- Негативные сценарии (деление на ноль, ввод букв вместо цифр, очень большие числа).
- Граничные значения (0,999, 1, 1,001).

3) Оформите тест-кейсы в виде таблицы:

| ID теста | Название                  | Шаги  | Ожидаемый результат      | Фактический результат | Статус (Pass/Fail) |
|----------|---------------------------|---|--------------------------|-----------------------|--------------------|
| ТС-001   | Сложение двух целых чисел | 1. Открыть калькулятор.<br>2. Нажать 2. 3. Нажать +.<br>4. Нажать 3. 5. Нажать =. | На экране отображается 5 | 5                     | Pass               |

Выполните все тест-кейсы и зафиксируйте фактический результат.

### 4. Результат (Отчет):

- 1) Таблица с 10 тест-кейсами и результатами их выполнения.
- 2) Вывод: какие функции работают корректно, а какие — с ошибками (если такие найдены).

## Практическая работа №2

Тема: «Интеграционное тестирование» (4 часа)

### 1. Цель работы:

Научиться проверять взаимодействие между отдельными модулями или компонентами программной системы.

### 2. Оснащение:

- Макет веб-приложения «Форма регистрации и входа» (можно использовать локальный учебный стенд или простой сайт с регистрацией, например, любой учебный проект).

- Браузер с инструментами разработчика (F12).

### 3.

Вам дана система, состоящая из трех модулей: Форма регистрации, Форма входа и База данных пользователей (имитация).

- 1) Проверьте сценарий «Регистрация нового пользователя → Вход в систему».
    - Зарегистрируйте нового пользователя (запомните логин/пароль).
    - Попробуйте войти в систему с этими данными.
    - Что проверяем: Передается ли созданная учетная запись из модуля регистрации в модуль входа?
  - 2) Проверьте сценарий «Вход с некорректными данными».
    - Попробуйте войти с неверным паролем.
    - Что проверяем: Корректно ли модуль входа обрабатывает и отклоняет неправильные данные из БД?
  - 3) Проверьте сценарий «Восстановление пароля» (если доступно).
  - 4) С помощью инструментов разработчика (вкладка Network) проследите, какие запросы отправляются на сервер при регистрации и входе.
4. Результат (Отчет):
- 1) Описание проверенных сценариев.
  - 2) Скриншоты успешной регистрации и входа.
  - 3) Вывод о корректности взаимодействия модулей (работает связка или есть ошибки).

### Практическая работа №3

Тема: «Тестирование безопасности»

#### 1. Цель работы:

Изучить основные уязвимости веб-приложений и научиться выполнять базовое тестирование безопасности.

#### 2. Оснащение:

– Специально подготовленный учебный сайт с уязвимостями (например, локальная копия старого учебного проекта) или тестовый полигон типа <http://testphp.vulnweb.com/> (Acunetix test site).

– Браузер.

#### 3. (выполнять только на разрешенных учебных полигонах!):

##### 1) Тест на SQL-инъекцию:

– Найдите на сайте форму поиска или страницу входа.

– Попробуйте ввести в поле для логина специальное значение: ' OR '1'='1 (или admin' --).

– Посмотрите, удастся ли обойти авторизацию или вызвать ошибку базы данных.

##### 2) Тест на XSS (межсайтовый скриптинг):

– Найдите на сайте поле ввода (комментарии, поиск).

– Попробуйте ввести туда простой скрипт: <script>alert('XSS')</script>.

– Если при загрузке страницы выскочило окно с предупреждением — уязвимость присутствует.

##### 3) Проверка протокола HTTPS:

– Определите, используется ли на сайте HTTPS.

– Если да, проверьте, корректно ли работает переадресация с HTTP на HTTPS.

4) Анализ заголовков безопасности:

– Откройте инструменты разработчика (F12) → вкладка Network → выберите любой запрос → посмотрите Response Headers.

– Найдите заголовки, связанные с безопасностью (например, Content-Security-Policy, X-Frame-Options). Если их нет, это недостаток.

4. Результат (Отчет):

1) Описание проведенных тестов.

2) Скриншоты результатов (например, всплывающее окно XSS).

3) Вывод об уровне безопасности протестированного ресурса и рекомендации по её повышению.

## Практическая работа №4

Тема: «Тестирование локализации и глобализации»

1. Цель работы:

Научиться проверять готовность программного продукта к использованию в разных странах (локализация интерфейса, поддержка языков, форматов дат, валют).

2. Оснащение:

– Любой веб-сайт или приложение с поддержкой нескольких языков (например, международные версии сайтов: Google, Wikipedia, онлайн-магазины).

– Браузер.

3.

1) Выберите сайт, имеющий как минимум две языковые версии (например, русскую и английскую).

2) Составьте список ключевых элементов интерфейса для проверки: меню, кнопки, заголовки, сообщения об ошибках, подписи полей.

3) Переключите интерфейс на русский язык. Проверьте:

– Все ли элементы интерфейса переведены? (Нет ли «англицизмов» или непере переведенных кусков).

– Корректно ли отображаются шрифты (особенно для кириллицы, если сайт заточен под латиницу).

4) Проверьте форматы (на примере вымышленных данных):

– Дата: Отображается ли дата в формате ДД.ММ.ГГГГ (для России) или ММ/ДД/ГГГГ (для США)?

– Числа: Правильно ли отображается десятичный разделитель (запятая или точка)?

– Валюта: Правильно ли отображается символ валюты (₽ или \$)?

5) Переключите язык на английский и повторите проверку.

4. Результат (Отчет):

1) Таблица с проверяемыми элементами и результатами проверки для каждого языка.

2) Скриншоты, иллюстрирующие найденные проблемы или подтверждающие корректную работу.

### 3) Вывод о качестве локализации.

#### Практическая работа №5

Тема: «Тестирование удобства использования (юзабилити-тестирование)»

##### 1. Цель работы:

Освоить методы оценки пользовательского опыта (UX), научиться выявлять проблемы, мешающие эффективной работе пользователя с интерфейсом.

##### 2. Оснащение:

– Прототип или готовый веб-сайт (можно взять разработанный в МДК.01.02 или любой другой на выбор).

– Секундомер (можно на телефоне).

– Два-три одnogруппника в роли «тестировщиков».

##### 3.

1) Выберите веб-сайт для тестирования (например, сайт библиотеки, интернет-магазин, сайт университета).

2) Сформулируйте 3-4 задачи (сценария) для пользователя, которые он должен выполнить на сайте. Задачи должны быть конкретными:

– Плохо: «Найди что-нибудь про Гарри Поттера».

– Хорошо: «Найди книгу 'Гарри Поттер и философский камень' и добавь её в корзину / закажи в библиотеке».

– Хорошо: «Найди контакты (телефон) организации и перейди на страницу 'О нас'».

3) Попросите одного из одnogруппников (испытуемого) выполнить эти задачи, не подсказывая ему. Включите секундомер. Наблюдайте за его действиями и записывайте:

– Время выполнения каждой задачи.

– Где он задумался или ошибся?

– Какие вопросы задает?

4) После выполнения задачи, коротко опросите испытуемого: было ли ему удобно, что показалось странным или непонятным.

5) Повторите тест с другим испытуемым.

##### 4. Результат (Отчет):

1) Список задач и среднее время их выполнения.

2) Описание наблюдений: проблемные места интерфейса, которые вызвали затруднения у пользователей.

3) Рекомендации по улучшению юзабилити (например: «Кнопку 'Заказать' нужно сделать более заметной», «Добавить поиск по сайту»).

#### Практическая работа №6

Тема: «Кросс-платформенное тестирование»

##### 1. Цель работы:

Научиться проверять корректность отображения и функциональности веб-приложения в различных браузерах и операционных системах.

##### 2. Оснащение:



- Любой веб-сайт или разработанный прототип.
- Доступ к разным браузерам: Chrome, Firefox, Edge (на Windows), Safari (если есть Mac).
- Эмуляторы мобильных устройств в браузере (инструменты разработчика).

3.

1) Выберите один и тот же веб-сайт (или свою страницу из ПР №6 по МДК.01.02).

2) Откройте его в трех разных браузерах (например, Chrome, Firefox, Edge) на одном компьютере. Сравните:

- Отображение шрифтов.
- Расположение кнопок и блоков (не «съехала» ли верстка?).
- Работу JavaScript (кликабельны ли кнопки, работают ли выпадающие списки).

3) Включите в браузере режим эмуляции мобильных устройств (F12 → Toggle device toolbar). Проверьте отображение сайта для iPhone 12 / Pixel 5 (или других популярных моделей). Есть ли горизонтальная прокрутка? Не налезает ли элементы друг на друга?

4) Составьте сводную таблицу.

Результат (Отчет):

1) Таблица вида:

| Браузер / Платформа         | Отображение (ОК/Не ОК) | Функциональность (ОК/Не ОК) | Замеченные проблемы               |
|-----------------------------|------------------------|-----------------------------|-----------------------------------|
| Google Chrome (Windows)     | ОК                     | ОК                          | —                                 |
| Mozilla Firefox (Windows)   | ОК                     | ОК                          | Мелкий шрифт в меню               |
| Microsoft Edge (Windows)    | Не ОК                  | ОК                          | Кнопка «Купить» наехала на фото   |
| Эмулятор iPhone 12 (Chrome) | Не ОК                  | Не проверялось              | Сайт не адаптирован, мелкий текст |

2. Вывод о кроссплатформенной совместимости ресурса.

## Практическая работа №7

Тема: «Выбрать технику тестирования и протестировать готовый веб-ресурс с использованием тест-планов. Создать отчет с результатами тестирования»

(Итоговая, комплексная работа)

1. Цель работы:

Выполнить полный цикл тестирования веб-ресурса, применив различные техники, разработав полный комплект тестовой документации и оформив итоговый отчет.

2. (Индивидуальный вариант):

Вам предоставлен готовый веб-ресурс (это может быть сайт-портфолио, интернет-магазин, блог, разработанный студентами старших курсов или найденный в интернете простой сервис). Ваша задача — провести его комплексное тестирование.

3. Этапы выполнения и состав проекта:

#### Этап 1. Анализ и планирование (1 час)

- Изучите предоставленный ресурс. Определите его основное назначение.
- Составьте Тест-план, который должен включать:
  - a) Объект тестирования (название, ссылка).
  - b) Виды тестирования, которые вы будете проводить (функциональное, UI, юзабилити, безопасности — хотя бы 3 вида).
  - c) График (последовательность) работ.
  - d) Критерии начала и окончания тестирования.

#### Этап 2. Проектирование тестов (1 час)

- На основе Тест-плана разработайте Чек-лист для быстрой проверки основных функций (например, «Главная страница загружается», «Работает поиск», «Есть контактная информация»).
- Разработайте 10-15 подробных тест-кейсов для наиболее критичных функций, покрывающих позитивные и негативные сценарии. Оформите их в табличном виде (как в ПР №1).

#### Этап 3. Выполнение тестирования (2 часа)

- Проведите тестирование согласно разработанным тест-кейсам и чек-листу.
- Задokumentируйте все найденные несоответствия и ошибки в виде Баг-репортов. Для каждого баг-репорта укажите:
  - a) ID, Краткое описание.
  - b) Шаги воспроизведения.
  - c) Фактический и ожидаемый результат.
  - d) Окружение (браузер, ОС).
  - e) Серьезность (критическая, значительная, незначительная).
  - f) Скриншот или видео с ошибкой.

#### Этап 4. Анализ результатов и подготовка отчета (2 часа)

- Проанализируйте результаты. На основе собранных данных составьте Итоговый отчет о тестировании (Test Summary Report). Отчет должен содержать:
  - a) Краткую информацию о проведенных работах.
  - b) Статистику: сколько тест-кейсов пройдено (Passed), сколько упало (Failed), сколько в блокировке (Blocked).
  - c) Список наиболее критичных найденных дефектов.
  - d) Оценку качества продукта (готов/не готов к релизу, требует доработки).
  - e) Рекомендации по улучшению.

#### 4. Результат (Отчет):

Один итоговый документ (или папка с файлами), включающий:

- 1) Тест-план.
- 2) Чек-лист.
- 3) Набор тест-кейсов.
- 4) Набор баг-репортов (хотя бы 3-5 шт.).
- 5) Итоговый отчет о тестировании.



## 2. Вопросы и для промежуточной аттестации

### МДК.01.01 Проектирование информационных ресурсов

1. Дайте определение понятиям «информационная система» и «информационный ресурс». В чем их различие?
2. Приведите классификацию информационных систем по различным признакам (по масштабу, по сфере применения, по способу организации).
3. Назовите и охарактеризуйте основные этапы проектирования информационных ресурсов.
4. Что такое системный анализ? Какие методы системного анализа применяются при проектировании ИС?
5. Раскройте понятие «концептуальная модель информационного ресурса». Для чего она нужна?
6. Какие графические нотации используются для моделирования бизнес-процессов? Дайте их краткую характеристику.
7. Что такое нотация BPMN? Назовите основные элементы BPMN-диаграммы.
8. Что такое UML? Какие диаграммы UML наиболее часто используются при проектировании веб-приложений?
9. Опишите назначение и состав диаграммы вариантов использования (Use Case Diagram).
10. Опишите назначение и состав диаграммы классов (Class Diagram).
11. Опишите назначение и состав диаграммы активности (Activity Diagram).
12. Опишите назначение и состав диаграммы последовательности (Sequence Diagram).
13. Что такое архитектура информационной системы? Перечислите основные типы архитектур (файл-серверная, клиент-серверная, двухуровневая, трехуровневая).
14. Раскройте понятия «функциональные требования» и «нефункциональные требования» к информационной системе.
15. Что такое техническое (ТЗ)? Какие разделы обязательно должны присутствовать в ТЗ?
16. Охарактеризуйте каскадную модель (Waterfall) разработки ПО. Назовите её преимущества и недостатки.
17. Охарактеризуйте итеративную и спиральную модели разработки ПО. В чем их отличие от каскадной модели?
18. Что такое гибкие методологии разработки (Agile)? Назовите их основные принципы.
19. Какие роли участников процесса проектирования вы знаете (заказчик, аналитик, архитектор, разработчик)?
20. Что такое CASE-средства? Приведите примеры CASE-средств для моделирования.
21. Как интерпретировать бизнес-требования заказчика для построения модели «как есть» (AS-IS) и «как должно быть» (TO-BE)?
22. Что такое методология IDEF0? Для чего применяются контекстная диаграмма и диаграммы декомпозиции?

23. Что такое репозиторий проекта? Какая информация в нем хранится на этапе проектирования?
24. Какие факторы влияют на выбор архитектуры для веб-приложения?
25. В чем заключается разница между логическим и физическим проектированием базы данных?
26. Что такое прототипирование требований? Какие виды прототипов существуют?
27. Как документировать результаты предпроектного обследования организации?
28. Какие стандарты регламентируют процессы проектирования и разработки ПО (обзор ГОСТ, ISO)?
29. Что такое риск в контексте проекта по разработке ПО? Какие риски могут возникнуть на этапе проектирования?
30. Перечислите критерии качества проектной документации.

#### МДК.01.02 Разработка интерфейсов пользователя

1. Дайте определение понятию «пользовательский интерфейс». Назовите основные типы интерфейсов (командный, графический, голосовой, жестовый).
2. Перечислите и кратко охарактеризуйте этапы проектирования пользовательского интерфейса.
3. Что такое UI (User Interface) и UX (User Experience)? В чем разница между этими понятиями?
4. Назовите основные принципы UX-дизайна (юзабилити).
5. Перечислите основные законы UX (закон Фиттса, закон Хика, закон Якоба и др.).
6. Что такое сайтмэп (Site Map)? Для чего он нужен и из каких элементов состоит?
7. Что такое вайрфрейм (Wireframe)? Какие бывают степени детализации вайрфреймов (low-fidelity, high-fidelity)?
8. Чем вайрфрейм отличается от макета (mockup) и прототипа (prototype)?
9. Какие существуют способы создания прототипов (бумажное прототипирование, цифровые инструменты)?
10. Какие инструменты (веб-сервисы и программы) используются для проектирования интерфейсов? Охарактеризуйте Figma.
11. Что такое фрейм (Frame) в Figma? Чем фрейм отличается от группы (Group)?
12. Что такое компоненты (Components) в Figma? Для чего они нужны? В чем разница между Main Component и Instance?
13. Что такое варианты компонентов (Component Variants)? Приведите пример использования.
14. Что такое Auto Layout в Figma? Какие задачи он решает?
15. Что такое привязки (Constraints) в Figma? Как они влияют на адаптивность макета?
16. Что такое сетка (Grid) и направляющие (Guides) в Figma? Для чего они используются?

17. Назовите основные элементы графического интерфейса (кнопки, поля ввода, переключатели, чекбоксы, выпадающие списки). Опишите их назначение.
18. Что такое гайдлайн (Guideline) или руководство по стилю? Какие разделы он должен содержать?
19. Что такое цветовая палитра? Какие существуют цветовые схемы и правила сочетания цветов в интерфейсе?
20. Как правильно подбирать шрифты для интерфейса? Что такое иерархия текста?
21. Что такое адаптивный (responsive) и отзывчивый дизайн? В чем разница?
22. Какие существуют подходы к разработке дизайна для мобильных устройств (Mobile First, Desktop First)?
23. Что такое модульная сетка? Как она помогает в верстке макетов?
24. Что такое интерактивный прототип? Как настроить связи (прототипирование) в Figma?
25. Какие правила проектирования форм (регистрации, обратной связи) необходимо соблюдать?
26. Как проводить анализ существующих интерфейсов (анализ конкурентов)? На что обращать внимание?
27. Что такое CJM (Customer Journey Map) и как она связана с проектированием интерфейсов?
28. Какие типичные ошибки допускают начинающие дизайнеры интерфейсов?
29. Какие современные тренды в UI/UX-дизайне вы знаете?
30. Как протестировать разработанный прототип на реальных пользователях и проанализировать ошибки?

#### МДК.01.03 Тестирование информационных ресурсов

1. Дайте определение понятию «тестирование программного обеспечения». Назовите цели тестирования.
2. Что такое качество ПО? Назовите основные характеристики качества по стандарту ISO/IEC 25010.
3. Перечислите принципы тестирования (поиск дефектов, невозможно провести исчерпывающее тестирование и др.).
4. Дайте классификацию видов тестирования по различным признакам (по уровню, по знанию системы, по степени автоматизации, по времени проведения).
5. Что такое функциональное тестирование? Что проверяется при его проведении?
6. Что такое тестирование пользовательского интерфейса (UI-тестирование)?
7. Что такое тестирование удобства использования (юзабилити-тестирование)?
8. Что такое тестирование безопасности? Какие основные уязвимости веб-приложений оно призвано выявить?
9. Что такое тестирование локализации и глобализации?
10. Что такое кросс-браузерное и кроссплатформенное тестирование?

11. Что такое интеграционное тестирование? Чем оно отличается от модульного?
12. В чем разница между статическим и динамическим тестированием?
13. В чем разница между тестированием методом «черного ящика», «белого ящика» и «серого ящика»?
14. Что такое тест-план (Test Plan)? Какие разделы он должен содержать?
15. Что такое тест-кейс (Test Case)? Какова его структура?
16. Что такое чек-лист (Checklist)? В каких случаях его удобнее использовать, чем тест-кейсы?
17. Что такое баг-репорт (Bug Report)? Какие поля обязательно должны быть в баг-репорте?
18. Что такое серьезность (Severity) и приоритет (Priority) дефекта? В чем разница?
19. Какие существуют техники тестирования (эквивалентное разделение, анализ граничных значений, попарное тестирование и др.)?
20. Что такое регрессионное тестирование? Когда оно проводится?
21. Что такое нагрузочное и стресс-тестирование? Для чего они нужны?
22. Какие существуют уровни тестирования (модульное, интеграционное, системное, приемочное)?
23. Что такое отчет о тестировании (Test Summary Report)? Какие выводы в нем делаются?
24. Перечислите популярные инструменты для автоматизации тестирования (Selenium, Postman, JMeter и др.).
25. Какие инструменты используются для отслеживания ошибок (Bug Tracking Systems)? (Jira, Redmine, YouTrack).
26. Что такое система контроля версий? Для чего она нужна в процессе разработки и тестирования?
27. Назовите основные команды Git (init, clone, add, commit, push, pull).
28. Что такое ветка (branch) в Git? Для чего используются ветки? Как происходит слияние веток (merge)?
29. Что такое конфликт версий в Git и как его разрешить?
30. Как организовать процесс тестирования при коллективной разработке с использованием системы контроля версий?

### 3. к экзамену по модулю

#### Теоретические вопросы:

1. Раскройте понятия «информационная система» и «информационный ресурс». Приведите классификацию информационных систем по масштабу и сфере применения.
2. Охарактеризуйте основные этапы жизненного цикла информационной системы согласно ГОСТ Р ИСО/МЭК 12207.
3. Назовите и подробно опишите этапы проектирования информационных ресурсов.
4. Что такое системный анализ? Какие методы системного анализа применяются на предпроектной стадии?

5. Раскройте понятие «концептуальная модель информационного ресурса». Какими средствами и нотациями она строится?
6. Сравните каскадную (Waterfall), итеративную и спиральную модели разработки ПО. Укажите их преимущества, недостатки и области применения.
7. Что такое гибкие методологии разработки (Agile, Scrum)? В чем их основные принципы и отличие от классических моделей?
8. Охарактеризуйте нотацию UML. Какие диаграммы UML используются при проектировании веб-приложений и для чего?
9. Опишите назначение, состав и правила построения диаграммы вариантов использования (Use Case Diagram).
10. Опишите назначение, состав и правила построения диаграммы классов (Class Diagram).
11. Опишите назначение, состав и правила построения диаграммы активности (Activity Diagram).
12. Охарактеризуйте нотацию BPMN. Назовите основные элементы BPMN-диаграммы и приведите пример моделирования бизнес-процесса.
13. Что такое архитектура информационной системы? Сравните файл-серверную, клиент-серверную и многоуровневую архитектуры.
14. Что такое техническое (ТЗ)? Перечислите основные разделы ТЗ на разработку информационного ресурса.
15. Как интерпретировать бизнес-требования заказчика? В чем разница между моделями AS-IS и TO-BE?
16. Что такое функциональные и нефункциональные требования? Приведите примеры.
17. Какие факторы влияют на выбор архитектуры веб-приложения?
18. Что такое CASE-средства? Приведите примеры инструментов для моделирования и их возможности.
19. Какие стандарты (ГОСТ, ISO) регламентируют процессы проектирования и документирования программного обеспечения?
20. Как оценить качество проектной документации? Назовите критерии.
21. Дайте определение понятиям UI (User Interface) и UX (User Experience). В чем заключается их взаимосвязь и различие?
22. Перечислите и охарактеризуйте основные этапы проектирования пользовательского интерфейса.
23. Назовите основные принципы и законы UX-дизайна (закон Фиттса, закон Хика, закон Якоба и др.).
24. Что такое сайтмэп (Site Map) и CJM (Customer Journey Map)? Для чего они нужны в проектировании?
25. Что такое вайрфрейм (Wireframe), макет (Mockup) и прототип (Prototype)? В чем их отличия?
26. Какие существуют способы и инструменты для создания прототипов? Охарактеризуйте возможности Figma.
27. Что такое компоненты и варианты компонентов в Figma? Для чего они используются?



28. Что такое Auto Layout и Constraints в Figma? Как они помогают в создании адаптивных интерфейсов?
29. Назовите основные элементы графического интерфейса и правила их применения.
30. Что такое гайдлайн (руководство по стилю)? Какие разделы он должен включать?
31. Какие правила подбора цветовой палитры и типографики в интерфейсах вы знаете?
32. Что такое адаптивный (responsive) и отзывчивый дизайн? В чем разница между подходами Desktop First и Mobile First?
33. Как проводится анализ существующих интерфейсов (анализ конкурентов)? На что обращать внимание?
34. Какие типичные ошибки допускаются при проектировании интерфейсов?
35. Как провести тестирование прототипа на пользователях и проанализировать полученные данные?
36. Дайте определение тестирования ПО. Назовите цели и задачи тестирования.
37. Что такое качество ПО? Охарактеризуйте модель качества ISO/IEC 25010.
38. Приведите классификацию видов тестирования по различным признакам (уровень, знание системы, степень автоматизации и др.).
39. В чем разница между функциональным и нефункциональным тестированием? Какие виды тестирования к ним относятся?
40. Охарактеризуйте тестирование методом «черного ящика», «белого ящика» и «серого ящика». Где каждый из них применяется?
41. Что такое тестирование безопасности? Перечислите основные виды уязвимостей веб-приложений.
42. Что такое интеграционное тестирование? Какие стратегии интеграции существуют (монолитная, нисходящая, восходящая)?
43. Что такое тестирование пользовательского интерфейса и юзабилити-тестирование? В чем их отличие?
44. Что такое регрессионное тестирование? Когда и для чего оно проводится?
45. Что такое нагрузочное, стресс-тестирование и тестирование стабильности?
46. Раскройте структуру и содержание тест-плана (Test Plan).
47. Что такое тест-кейс (Test Case) и чек-лист (Checklist)? Какова структура тест-кейса?
48. Что такое баг-репорт (Bug Report)? Опишите обязательные поля баг-репорта и правила его оформления.
49. В чем разница между серьезностью (Severity) и приоритетом (Priority) дефекта? Приведите примеры.
50. Какие техники тестирования (эквивалентное разделение, анализ граничных значений, попарное тестирование) вы знаете? Приведите примеры их применения.
51. Что такое система контроля версий (VCS)? Для чего она необходима в коллективной разработке?

52. Охарактеризуйте Git как систему контроля версий. Назовите основные команды Git (init, clone, add, commit, push, pull, status).

53. Что такое ветка (branch) в Git? Как создать ветку, переключаться между ветками и выполнять слияние (merge)?

54. Что такое конфликт версий в Git? Каковы причины его возникновения и алгоритм разрешения?

55. Что такое удаленные репозитории (GitHub, GitLab)? Как организовать совместную работу над проектом с их использованием?

56. Что такое интеграция программного кода? Какие задачи решаются на этапе интеграции?

57. Что такое CI/CD (Continuous Integration / Continuous Deployment)? Каковы основные принципы?

Практико-ориентированные задания:

1. Проведите анализ предметной области для интернет-магазина по продаже спортивных товаров. Выделите не менее трех основных бизнес-процессов, опишите их текстово и постройте модель AS-IS в нотации Activity Diagram.

2. На основе требований заказчика: *«Нужен сайт для фотостудии, где клиенты могут просматривать портфолио, узнавать цены и записываться на фотосессию»* — составьте фрагмент технического задания (разделы «Основание для разработки» и «Требования к функциям»).

3. Постройте диаграмму вариантов использования (Use Case) для системы онлайн-бронирования отелей. Выделите актеров (гость, зарегистрированный пользователь, администратор) и основные варианты использования.

4. Разработайте концептуальную модель базы данных для системы учета студентов (ФИО, группа, оценки по предметам) в виде диаграммы классов UML.

5. Для процесса «Оформление заказа в интернет-магазине» постройте диаграмму активности (Activity Diagram), отражающую все возможные пути (успешная оплата, отказ от заказа, ошибка при оплате).

6. Сравните каскадную и итеративную модели разработки для проекта по созданию корпоративного сайта. Обоснуйте, какую модель вы выберете и почему.

7. Проведите декомпозицию бизнес-процесса «Прием пациента в поликлинике» с использованием нотации IDEF0 (контекстная диаграмма и первый уровень декомпозиции — не менее 3-х блоков).

8. Составьте глоссарий проекта (не менее 10 терминов) для предметной области «Система управления задачами (таск-трекер)».

9. Разработайте макет экранных форм (desktop версия) для страницы «Профиль пользователя» в личном кабинете фитнес-приложения. Используйте любой графический редактор или бумажное прототипирование с фото.

10. Проанализируйте нефункциональные требования для мобильного приложения банка. Предложите не менее 5 требований к производительности, безопасности и удобству использования.

11. В Figma создайте компонент «Карточка новости» (изображение, заголовок, дата, краткое описание). Используйте Auto Layout для вертикального расположения элементов и настройте отступы.

12. Разработайте низкодетализированный прототип (lo-fi) главной страницы интернет-магазина книг. Отобразите расположение шапки, поиска, каталога, блока с рекомендациями и подвала.

13. На основе созданного в задании 2.2 прототипа разработайте высокодетализированный макет (hi-fi) одной секции (например, блока с рекомендациями книг) с применением цветовой схемы и типографики.

14. Создайте в Figma адаптивный компонент «Кнопка с иконкой». Используйте варианты (variants) для состояний Default, Hover, Pressed и Disabled.

15. Разработайте сайтмэп для веб-приложения «Блог о путешествиях». Включите не менее 10 страниц, сгруппированных по разделам.

16. Составьте гайдлайн (UI-kit) для проекта, включающий: цветовую палитру (основные, дополнительные, семантические цвета), типографику (заголовки, текст), стили кнопок и полей ввода.

17. Создайте интерактивный прототип в Figma для сценария «Регистрация нового пользователя» (3 экрана: форма регистрации, подтверждение email, приветственный экран). Настройте связи между элементами.

18. Проведите анализ юзабилити существующего сайта (например, сайта вашего колледжа/университета). Найдите не менее 3 проблем и предложите пути их решения.

19. Разработайте CJM (карту пути пользователя) для сценария «Покупка товара в мобильном приложении». Опишите этапы, действия пользователя, его эмоции и точки контакта.

20. Адаптируйте разработанный вами ранее макет главной страницы (из задания 2.2) под мобильное устройство (ширина экрана 375px). Используйте Auto Layout для перестроения блоков.

21. Для функции авторизации (логин и пароль) разработайте 5 тест-кейсов, покрывающих позитивные и негативные сценарии (включая граничные значения и некорректные данные).

22. При тестировании формы обратной связи обнаружено, что после отправки сообщения пользователь не получает уведомление об успешной отправке. Оформите баг-репорт с подробным описанием.

23. Составьте чек-лист для тестирования страницы «Контакты» (наличие карты, корректность телефона/email, работа формы обратной связи). Минимум 10 пунктов.

24. Проведите функциональное тестирование калькулятора на сайте (или в ОС) и заполните таблицу: ID теста, входные данные, ожидаемый результат, фактический результат, статус.

25. Разработайте тест-план для модуля «Поиск товаров» в интернет-магазине. Включите цели, объекты тестирования, виды тестирования и критерии начала/окончания.

26. Проведите тестирование локализации для сайта с переключением языков (например, русский и английский). Проверьте перевод основных элементов и составьте отчет о найденных проблемах.



27. Выполните кросс-браузерное тестирование главной страницы любого сайта в трех браузерах (Chrome, Firefox, Edge). Составьте сравнительную таблицу с описанием различий (если они есть).

28. Составьте итоговый отчет о тестировании (Test Summary Report) по результатам выполнения 10 тест-кейсов, из которых 7 прошли успешно, 2 упали, 1 был заблокирован.

29. Придумайте примеры дефектов с разной серьезностью (блокирующий, критический, значительный, незначительный, тривиальный) и приоритетом (высокий, средний, низкий). Обоснуйте ваш выбор.

30. Для поля ввода «Возраст» (допустимые значения от 18 до 100) примените технику анализа граничных значений и составьте набор тестовых данных.

31. Опишите последовательность команд Git для создания нового репозитория, добавления файла, создания коммита и отправки изменений в удаленный репозиторий на GitHub.

32. Смоделируйте ситуацию конфликта при слиянии веток в Git. Опишите пошагово, как его обнаружить и разрешить.

33. Представьте, что вы работаете в команде над проектом. Опишите стратегию работы с ветками (git flow), которую вы предложите: какие ветки будут основными, для чего они предназначены.

34. Создайте в Figma (или на бумаге) схему, отражающую процесс интеграции нового модуля в существующую систему. Укажите этапы: разработка, тестирование, слияние, развертывание.

35. Опишите порядок действий при обнаружении критической ошибки в коде, который уже находится в основной ветке (main/master). Как исправить ошибку с использованием Git?