

Министерство цифрового развития, связи и массовых коммуникаций Российской Федерации  
Федеральное государственное бюджетное образовательное учреждение высшего образования  
«Сибирский государственный университет телекоммуникаций и информатики»  
(СибГУТИ)  
Уральский технический институт связи и информатики (филиал) в г. Екатеринбурге  
(УрТИСИ СибГУТИ)

УТВЕРЖДАЮ  
Директор УрТИСИ СибГУТИ  
Минина Е.А.  
« 28 » 11 2025 г.

## ОЦЕНОЧНЫЕ СРЕДСТВА ТЕКУЩЕГО КОНТРОЛЯ И ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ

### ПО ДИСЦИПЛИНЕ Б1.В.01 Web-технологии

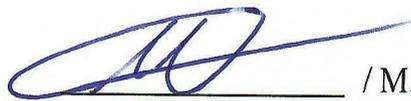
Направление подготовки / специальность: **09.03.01 «Информатика и  
вычислительная техника»**

Направленность (профиль) /специализация: **Инженерия программного  
обеспечения и искусственного интеллекта**

Форма обучения: **очная**

Год набора: 2026

Разработчик (-и):  
ст.преподаватель

 / М.Ю. Казанцев /  
подпись

к.т.н., доцент

 / Т.А. Черных /  
подпись

Оценочные средства обсуждены и утверждены на заседании информационных систем и технологий (ИСТ)

Протокол от 27.11.2025 г. № 3

Заведующий кафедрой  / Д.И. Бурумбаев /  
подпись

Екатеринбург, 2025

Министерство цифрового развития, связи и массовых коммуникаций Российской Федерации  
Федеральное государственное бюджетное образовательное учреждение высшего образования  
«Сибирский государственный университет телекоммуникаций и информатики»  
(СибГУТИ)  
Уральский технический институт связи и информатики (филиал) в г. Екатеринбурге  
(УрТИСИ СибГУТИ)

УТВЕРЖДАЮ  
Директор УрТИСИ СибГУТИ  
\_\_\_\_\_ Минина Е.А.  
« \_\_\_\_ » \_\_\_\_\_ 2025 г.

## ОЦЕНОЧНЫЕ СРЕДСТВА ТЕКУЩЕГО КОНТРОЛЯ И ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ

### ПО ДИСЦИПЛИНЕ Б1.В.01 Web-технологии

Направление подготовки / специальность: **09.03.01 «Информатика и  
вычислительная техника»**

Направленность (профиль) /специализация: **Инженерия программного  
обеспечения и искусственного интеллекта**

Форма обучения: **очная**

Год набора: 2026

Разработчик (-и):  
ст.преподаватель

\_\_\_\_\_ / М.Ю. Казанцев /  
подпись

к.т.н., доцент

\_\_\_\_\_ / Т.А. Черных /  
подпись

Оценочные средства обсуждены и утверждены на заседании информационных систем и технологий (ИСТ)

Протокол от 27.11.2025 г. № 3

Заведующий кафедрой \_\_\_\_\_ / Д.И. Бурумбаев /  
подпись

Екатеринбург, 2025

## 1. Перечень компетенций и индикаторов их достижения

Процесс изучения дисциплины направлен на формирование следующих компетенций:

Код и наименование компетенции	Код и наименование индикатора достижения компетенций	Этап	Предшествующие этапы (с указанием дисциплин/практик)
ПК 1 Способен проектировать и разрабатывать программное обеспечение	ПК 1.1 Знает современные методы, средства и стандарты для проектирования и разработки программного обеспечения ПК 1.2 Умеет применять современные технологии для проектирования и разработки программного обеспечения ПК 1.3 Владеет навыками проектирования и разработки программного обеспечения ПК 1.4 Знает методы, средства и стандарты проектирования баз данных ПК 1.5 Умеет применять методы и средства проектирования баз данных ПК 1.6 Владеет навыками использования методов и средств проектирования баз данных ПК 1.7 Знает методы и средства дизайна и проектирования программных интерфейсов ПК 1.8 Владеет навыками дизайна и проектирования программных интерфейсов	1	-

Форма итоговой аттестации по дисциплине – экзамен

## 2. Показатели, критерии и шкалы оценивания компетенций

2.1 Показателем оценивания компетенций на этапе их формирования при изучении дисциплины является уровень их освоения.

Индикатор освоения компетенции	Показатель оценивания	Критерий оценивания
ПК 1.1 Знает современные методы, средства и стандарты для проектирования и разработки	Тест/опрос по тематике web-разработки: архитектура клиент–сервер, HTTP/HTTPS, REST, основы web-безопасности (CORS, CSRF, XSS — на уровне понимания),	не менее 70% верных ответов; корректно объясняет базовые понятия (HTTP-методы/коды, REST, отличие SSR/SPA на

программного обеспечения	HTML5/CSS3, JS/TS (на уровне терминов), принципы модульности и SOLID (в применении к web), Git, основы тестирования и сборки	уровне курса), знает назначение Git и инструментов сборки/тестирования
ПК 1.2 Умеет применять современные технологии для проектирования и разработки программного обеспечения	Выполнение практических занятий: разработка frontend (SPA/SSR), backend API, работа с маршрутизацией, валидацией, обработкой ошибок, использование Git, подключение зависимостей, сборка проекта, настройка окружения (.env), базовая безопасность (валидация, CORS)	приложение собирается и запускается; требования ТЗ реализованы; корректная структура проекта; запросы/ответы API соответствуют контракту; нет критических уязвимостей уровня «вставка сырого SQL/отсутствие валидации»
ПК 1.3 Владеет навыками проектирования и разработки программного обеспечения	Итоговый мини-проект в рамках практических заданий: полноценное web-приложение (frontend+backend), демонстрация архитектуры, тестов и документации; работа с репозиторием (ветки/коммиты)	проект функционально завершен и стабильно работает; архитектура обоснована (слои/модули); есть README и инструкция запуска; реализованы базовые сценарии; история Git отражает процесс разработки; есть минимум базовых тестов или проверок
ПК 1.4 Знает методы, средства и стандарты проектирования баз данных	Тест/опрос: ER-моделирование, нормализация, ключи и ограничения, транзакции, индексы (на базовом уровне), связь web-приложения с БД, миграции	не менее 70%; корректно объясняет назначение PK/FK/UNIQUE, нормализацию (до 3НФ), зачем нужны миграции и транзакции
ПК 1.5 Умеет применять методы и средства проектирования баз данных	Практические занятия: проектирование схемы под web-кейс, написание DDL, настройка ограничений, миграции, написание запросов для типовых операций (CRUD/поиск/фильтры)	схема соответствует требованиям; DDL/миграции применяются без ошибок; корректно заданы ключи и ограничения; запросы возвращают корректные данные на тестовых наборах
ПК 1.6 Владеет навыками использования методов и средств проектирования баз данных	Мини-проект в рамках практических занятий: интеграция приложения с БД через ORM/SQL-клиент, миграции, транзакционные сценарии, оптимизация 1–2 запросов (индекс/план)	интеграция работает стабильно; миграции воспроизводимы; транзакции применены там, где нужно; нет SQL-инъекций (параметризация/ORM); показан эффект индекса или обоснование

ПК 1.7 Знает методы и средства дизайна и проектирования программных интерфейсов	Тест/опрос: основы UX/UI, пользовательские сценарии, wireframes, дизайн-системы/компоненты, принципы доступности (a11y на базовом уровне), адаптивность, паттерны навигации	не менее 70%; корректно объясняет цели wireframe/прототипа, принципы консистентности, базовые требования доступности и адаптивной верстки
ПК 1.8 Владеет навыками дизайна и проектирования программных интерфейсов	Работа в рамках практических занятий: разработка макетов (Figma или аналог), реализация адаптивного интерфейса, компоненты, формы и валидация, пользовательские сценарии	макеты/интерфейс соответствуют сценариям; интерфейс адаптивен; единый стиль компонентов; формы удобны и валидируются; соблюдены базовые принципы UX (понятная навигация, обратная связь, читаемость)

### Шкала оценивания.

#### Экзамен

5-балльная шкала	Критерии оценки
«отлично»	На экзаменационные вопросы даны полные аргументированные ответы. Студент демонстрирует сформированность компетенций ПК 1.1–ПК 1.8 на итоговом уровне: глубокое понимание клиент-серверной архитектуры, HTTP/HTTPS, REST, базовых вопросов безопасности (валидация, XSS/CSRF/CORS на уровне принципов), уверенное владение HTML/CSS/JS (или TS), принципами проектирования и структурирования кода, инструментами сборки и контроля версий. Уверенно проектирует и реализует web-приложение: описывает структуру frontend/backend, разрабатывает API, корректно работает с базой данных (ER-модель, ключи, ограничения, миграции, транзакции), демонстрирует навыки проектирования интерфейсов (сценарии, макеты, адаптивность). Практические задачи решаются без принципиальных ошибок, решения обоснованы.
«хорошо»	На экзаменационные вопросы даны в целом полные ответы, но с незначительными неточностями или замечаниями преподавателя. Компетенции сформированы на среднем уровне: основные понятия web-разработки и БД усвоены, но допускаются отдельные ошибки в терминологии, архитектурных решениях, деталях HTTP/REST, безопасности, проектировании схемы или интерфейса. Практическая задача выполнена, но есть недочеты (частично неверная валидация, недостаточная обработка ошибок, слабая декомпозиция, мелкие ошибки в запросах/миграциях), исправляемые после замечаний.
«удовлетворительно»	Ответы даны со слабой аргументацией, студент отвечает неуверенно, требуется много наводящих вопросов. Компетенции сформированы на базовом уровне: знание материала фрагментарное, есть пробелы по ключевым темам (HTTP/REST, безопасность, проектирование БД, миграции, проектирование интерфейса). В практической задаче допускаются существенные

	ошибки (неверный выбор подхода, логические ошибки в запросах, отсутствие валидации/обработки ошибок), но после подсказок студент способен частично исправить решение.
«неудовлетворительно»	Компетенции сформированы на уровне ниже порогового: отсутствуют базовые знания по web-разработке и проектированию БД/интерфейсов. Студент не может объяснить основные принципы (HTTP/REST, клиент-сервер, валидация, ключи/ограничения, миграции, базовые принципы UI/UX) и не способен выполнить практическую задачу. Дисциплинарные компетенции не сформированы.

### 3. Методические материалы, определяющие процедуры оценивания по дисциплине

#### 3.1. В ходе реализации дисциплины используются следующие формы и методы текущего контроля

Тема и/или раздел	Формы/методы текущего контроля успеваемости
ПК 1 Способен проектировать и разрабатывать программное обеспечение	
Введение в веб и инструменты разработчика	Самостоятельная работа, конспект лекции
HTML и семантическая верстка	Самостоятельная работа, конспект лекции
CSS основы и адаптивная верстка	Самостоятельная работа, конспект лекции
JavaScript в браузере основы языка и DOM	Самостоятельная работа, конспект лекции
Асинхронность и работа с API	Самостоятельная работа, конспект лекции
Node.js основы серверной разработки	Самостоятельная работа, конспект лекции
Проектирование REST API и безопасность веб приложения	Самостоятельная работа, конспект лекции
Базы данных для веб приложения и работа из Node.js	Самостоятельная работа, конспект лекции
Современный фронтенд на фреймворке и итоговый проект	Самостоятельная работа, конспект лекции
Настройка окружения веб разработчика, Node.js npm, VS Code, Git, запуск первого проекта	Практическая работа
Семантическая HTML страница, формы, базовая доступность и валидация	Практическая работа
Адаптивная верстка макета, Flexbox Grid, mobile first, организация CSS	Практическая работа
Интерактивность на JS, работа с DOM, события, валидация форм на клиенте	Практическая работа
Запросы к внешнему API через fetch, обработка ошибок, вывод данных на страницу	Практическая работа
Мини проект фронтенд виджет, асинхронность async await, состояния загрузки и ошибок	Практическая работа

Создание backend на Express, маршруты middleware, обработка ошибок, структура проекта	Практическая работа
Проектирование и реализация REST API CRUD, статус коды, пагинация, базовая авторизация	Практическая работа
Защита web приложения, хранение секретов, rate limiting, защита от типовых атак на ввод	Практическая работа
Проектирование схемы БД и реализация в PostgreSQL, таблицы связи индексы	Практическая работа
Интеграция Node.js с PostgreSQL, параметризованные запросы, миграции, транзакции на базовом уровне	Практическая работа
Итоговый проект, SPA интерфейс плюс REST API плюс БД, деплой и демонстрация результата	Практическая работа

### 3.2. Типовые материалы текущего контроля успеваемости обучающихся

#### ПК 1 Способен проектировать и разрабатывать программное обеспечение.

Пример задания на практическое занятие

Цель: Закрепить навыки разработки web-приложения по схеме frontend + backend + база данных, включая базовую безопасность и деплой.

Задание: Разработка мини-приложения «Список задач» (SPA + REST API + PostgreSQL)

Задачи: Определение требований:

Функциональные: регистрация/вход (упрощенно), создание/редактирование/удаление задач, изменение статуса, фильтрация и пагинация списка.

Нефункциональные: валидация данных на клиенте и сервере, обработка ошибок единым форматом, хранение секретов через .env, защита от SQL-инъекций (параметризованные запросы).

Проектирование:

Спроектируйте структуру страниц SPA (список задач, форма задачи, страница входа).

Опишите контракт REST API (эндпоинты CRUD, коды ответов, схема ошибок).

Спроектируйте схему БД (таблицы users/tasks, ключи, связи, индексы).

Реализация:

Создайте backend на Express: маршруты, middleware, обработка ошибок.

Реализуйте REST API (CRUD), фильтрацию/пагинацию.

Подключите PostgreSQL, выполните миграции, используйте параметризованные запросы, добавьте транзакцию для составной операции (если есть).

Создайте frontend (SPA) и подключите к API через fetch/async-await, реализуйте состояния загрузки и ошибок.

Тестирование и отладка:

Протестируйте основные сценарии (создание/изменение/удаление, ошибки валидации, недоступность API).

Проверьте, что ввод не приводит к выполнению SQL-кода (параметризация).

Документация:

Подготовьте README: установка, запуск, переменные окружения, миграции, примеры запросов.

Внедрение:

Соберите проект и разверните на тестовом сервере (локально через docker compose или на VPS).

Отчет:

Подготовьте отчет, который включает:

Описание требований и архитектуры.

Контракт API и схему БД.

Ссылку на репозиторий и инструкцию запуска.

Скриншоты работы приложения и выводы.

Типовые вопросы и задания к экзамену

1. Как работает веб: клиент–сервер, роль HTTP/HTTPS, DNS.
2. HTTP методы и коды ответов: GET/POST/PUT/PATCH/DELETE, 200/201/204/400/401/403/404/409/500.
3. В чем отличие REST от RPC (на базовом уровне)?
4. Что такое CORS и почему возникает ошибка CORS?
5. Асинхронность в JS: Promise, async/await, обработка ошибок.
6. Семантическая верстка: зачем нужны семантические теги и базовые принципы доступности.
7. Адаптивная верстка: mobile first, Flexbox vs Grid.
8. Express: маршруты, middleware, обработка ошибок.
9. Валидация входных данных на клиенте и на сервере: зачем нужны обе.
10. Базовые угрозы веб-приложений: SQL-инъекции, XSS, CSRF (на уровне идеи) и меры защиты.
11. Проектирование БД для веб-приложения: ключи, связи, индексы, миграции.
12. Параметризованные запросы: зачем нужны и как защищают от SQL-инъекций.
13. Транзакции: когда применять и что такое атомарность.
14. Архитектура проекта: разделение frontend/backend, структура директорий, конфигурация через .env.
15. Практическое задание: спроектировать REST API и схему БД для сущности и написать пример запроса fetch + обработку ошибок.

Банк контрольных вопросов, заданий и иных материалов, используемых в процессе процедур текущего контроля и промежуточной аттестации находится в учебно-методическом комплексе дисциплины и/или представлен в электронной информационно-образовательной среде по URI: <http://www.aup.uisi.ru>.

### **3.3. Методические материалы проведения текущего контроля и промежуточной аттестации обучающихся**

Перечень методических материалов для подготовки к текущему контролю и промежуточной аттестации:

1. Методические указания по выполнению практических занятий по дисциплине «Технологии командной разработки программного обеспечения». –URL: <http://aup.uisi.ru/4629963/>
- 2 Образовательная среда УрТИСИ СибГУТИ – URL: <https://moodle.uisi.ru>