Федеральное государственное бюджетное образовательное

учреждение высшего образования

«Алтайский государственный технический университет

им. И.И. Ползунова»

Факультет Информационных технологий

Кафедра Прикладная математика

Отчет защищен с оценкой\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Я.Б.Еремин

(подпись преподавателя) (инициалы, фамилия)

“\_\_\_\_”\_\_\_\_\_\_\_\_\_\_\_ 2022 г.

Отчет

по лабораторной работе №2

Багрепорты и юнит-тесты

по дисциплине «**Верификация и тестирование программного обеспечения**»

ЛР 09.03.04.00.10 ОТ

(обозначение документа)

Студент группы ПИ-82 A.А.Бобров

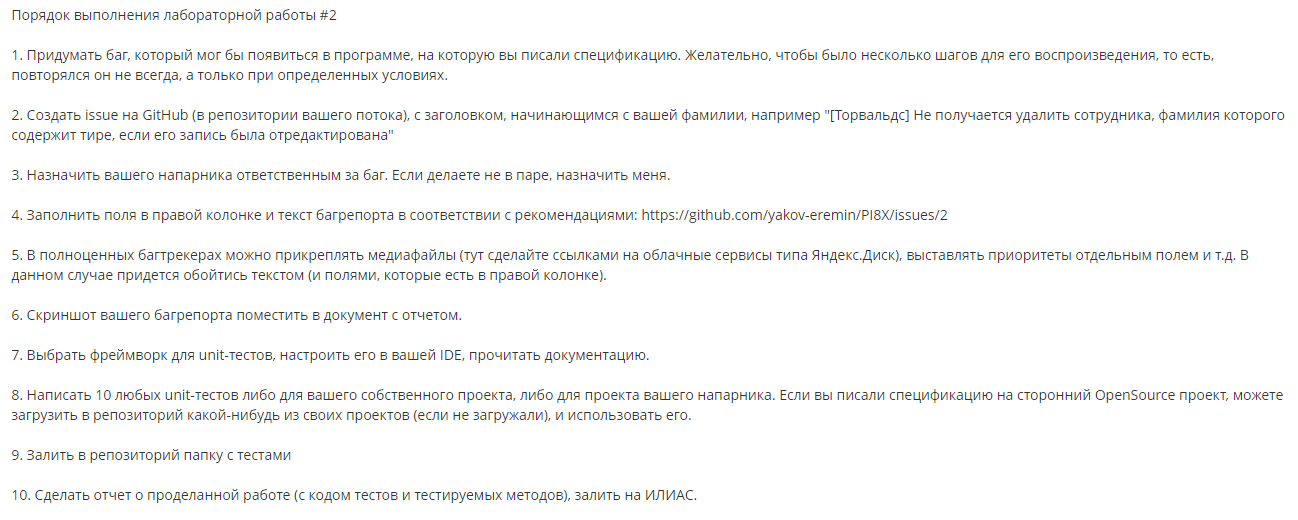
(инициалы, фамилия)

Преподаватель старший преподаватель Я.Б.Еремин

(должность, ученое звание) (инициалы, фамилия)

Барнаул 2022

**Задание:**



**Ход выполнения:**

Баг был выбран для работы Реутова Антона и оформлен в виде issue на GitHub. Для второй части работы был выбран проект на NodeJS, в котором было произведено unit-тестирования при помощи фреймворка Mocha.

Для установки фреймворка необходимо в директории проекта прописать команду:

npm –i mocha –save-dev

Далее будет произведена установка, после которой необходимо вписать в файл package.json команду запуска тестов:

"scripts": {

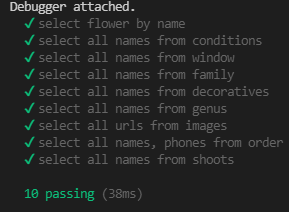
"test": "mocha \*.test.js"

},

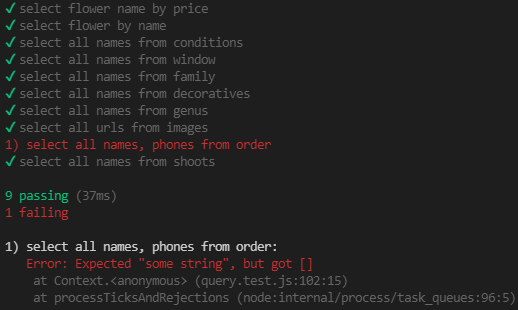
Таким образом все тесты будут запускаться разом по команде:

npm test

Пример вывода в консоль результатов тестирования проекта



При неудачном тесте будет отображены следующие данные:



Тестируемые функции:

// Получение всех строк из таблицы

selectAllFromTable = async (tableName) => {

    let result = await pool.query(`SELECT \* FROM ${tableName}`);

    return result[0];

};

// Получение конкретных строк из таблицы с условием

selectFromTable = async (tableName, value, where) => {

    let result;

    if(where)

        result = await pool.query(`SELECT ${value} FROM ${tableName} WHERE ${where}`);

    else

        result = await pool.query(`SELECT ${value} FROM ${tableName}`);

    return result[0];

};

Код тестов:

const Query = require('./query');

it('select flower name by price', async () => {

    var expectedResult = [{name: 'Эхинокактус Грузона'}];

    const result = await Query.selectFromTable("flowers", "name", "price=7700");

    if (JSON.stringify(expectedResult) !== JSON.stringify(result)) {

        throw new Error(

            `Expected ${JSON.stringify(expectedResult)}, but got ${JSON.stringify(result)}`

        )

    }

});

it('select flower by name', async () => {

    var expectedResult = [

        {

            "PK\_Flowers":12,

            "name":"Эхинокактус Грузона",

            "price":7700,

            "description":"Растение, распространённое в Мексике; популярное комнатное растение.",

            "PK\_Genus":3,

            "PK\_Decoratives":3,

            "PK\_Shoots":1,

            "PK\_Window":1

        }

    ];

    const result = await Query.selectFromTable("flowers", "\*", `name="Эхинокактус Грузона"`);

    if (JSON.stringify(expectedResult) !== JSON.stringify(result)) {

        throw new Error(

            `Expected ${JSON.stringify(expectedResult)}, but got ${JSON.stringify(result)}`

        )

    }

});

it('select all names from conditions', async () => {

    var expectedResult = [

        {"name":"Теневыносливые"},

        {"name":"Светолюбивые"},

        {"name":"Влаголюбивые"},

        {"name":"Засухоустойчивые"}

    ];

    const result = await Query.selectFromTable("conditions", "name");

    if (JSON.stringify(expectedResult) !== JSON.stringify(result)) {

        throw new Error(

            `Expected ${JSON.stringify(expectedResult)}, but got ${JSON.stringify(result)}`

        )

    }

});

it('select all names from window', async () => {

    var expectedResult = [{"name":"Западное"},{"name":"Восточное"}];

    const result = await Query.selectFromTable("window", "name");

    if (JSON.stringify(expectedResult) !== JSON.stringify(result)) {

        throw new Error(

            `Expected ${JSON.stringify(expectedResult)}, but got ${JSON.stringify(result)}`

        )

    }

});

it('select all names from family', async () => {

    var expectedResult = [{"name":"Ароидные"},{"name":"Новое семейство"},{"name":"Кактусовые"}];

    const result = await Query.selectFromTable("family", "name");

    if (JSON.stringify(expectedResult) !== JSON.stringify(result)) {

        throw new Error(

            `Expected ${JSON.stringify(expectedResult)}, but got ${JSON.stringify(result)}`

        )

    }

});

it('select all names from decoratives', async () => {

    var expectedResult = [{"name":"Лиственные"},{"name":"Цветущие"},{"name":"Суккуленты"},{"name":"Кактусы"}];

    const result = await Query.selectFromTable("decoratives", "name");

    if (JSON.stringify(expectedResult) !== JSON.stringify(result)) {

        throw new Error(

            `Expected ${JSON.stringify(expectedResult)}, but got ${JSON.stringify(result)}`

        )

    }

});

it('select all names from genus', async () => {

    var expectedResult = [{"name":"Замиокулькас"},{"name":"Новый род"},{"name":"Эхинокактус"}];

    const result = await Query.selectFromTable("genus", "name");

    if (JSON.stringify(expectedResult) !== JSON.stringify(result)) {

        throw new Error(

            `Expected ${JSON.stringify(expectedResult)}, but got ${JSON.stringify(result)}`

        )

    }

});

it('select all urls from images', async () => {

    var expectedResult =  [{"url":"img1-1647248337697.jpg"},{"url":"img2-1647248337698.jpg"},{"url":"img3-1647248337701.jpg"}];

    const result = await Query.selectFromTable("images", "url");

    if (JSON.stringify(expectedResult) !== JSON.stringify(result)) {

        throw new Error(

            `Expected ${JSON.stringify(expectedResult)}, but got ${JSON.stringify(result)}`

        )

    }

});

it('select all names, phones from order', async () => {

    var expectedResult = [];

    const result = await Query.selectFromTable("\`order\`", "name, number");

    if (JSON.stringify(expectedResult) !== JSON.stringify(result)) {

        throw new Error(

            `Expected ${JSON.stringify(expectedResult)}, but got ${JSON.stringify(result)}`

        )

    }

});

it('select all names from shoots', async () => {

    var expectedResult = [{"name":"Прямостоячие"},{"name":"Вьющиеся"},{"name":"Кустовидные"},{"name":"Розетки"}];

    const result = await Query.selectFromTable("shoots", "name");

    if (JSON.stringify(expectedResult) !== JSON.stringify(result)) {

        throw new Error(

            `Expected ${JSON.stringify(expectedResult)}, but got ${JSON.stringify(result)}`

        )

    }

});