JUDEX

Проект по курсу «Компьютерные технологии»

(инструкция по применению)

Преподаватель:

Сергей Леонидович Бабичев

Члены команды:

Игорь Тараймович, 777 Кирилл Пучков, 777 Дмитрий Гущин, 772

Оглавление

Ι	Техническое задание	3
1	Введение	4
	Назначение	4
	Область действия	4
	Определения, акронимы и сокращения	4
	Ссылки	4
2	Общее описание	5
	Взаимодействие продукта	5
	Функции продукта	5
	Ограничения	5
	Допущения и зависимости	5
3	Детальные требования	6
	Требования к внешним интерфейсам	6
	Функциональные требования	6
	Требования к производительности	6
TT	Отчет	7
11	Oraci	•
4	Обзор проекта	8
	Используемые языки	8
	Структура проекта	9
	Ход работы	10
5	Подробности реализации	13
	Описание функций	13

Часть I Техническое задание

Введение

Назначение
Этот проект — семестровый (4 семестр) обязательный проект на курсе
«Компьютерные технологии».
Целью нашей работы было создание автоматической системы проверки
решений контестов по программированию.
Область действия
Целевой аудиторией нашего проекта были и остаются студенты или школь ники, увлекающиеся программирование. В частности заинтересованы в продукте те, кто имеет стремление решать контесты.
Определения, акронимы и сокращения
Judex~(от латин. $)$ — судья. САПП (сокр.) — система автоматической проверки программ.
Ссылки
Сайт проекта http://judex.vdi.mipt.ru

Общее описание

Взаимодействие продукта
Мы использовали Mongo DB Driver для подключения к базе данных на одноименном сервере https://github.com/mongodb/mongo-go-driver/ Для безопасного подключения к серверу мы использовали reCAPTCHA https://github.com/dpapathanasiou/go-recaptcha
Функции продукта
Функции продукта следующие:
• Хранение задач.
• Отправка на автоматизированную проверку.
• Возможность узнать результат.
Ограничения
Компиляции файлов, написанных на языках ${\bf C}$ и ${\bf C}++.$
Допущения и зависимости
Отсутствует.

Детальные требования

	Гребования к внешним интерфейсам
Интерфеі	йсы пользователя
Браузер с по	ддержкой HTML5 и разрешенными скриптами.
Интерфеі Нет.	йсы аппаратного обеспечения
Интерфеі Нет.	йсы программного обеспечения
Интерфей	йсы взаимодействия
Подключени	е к интернету.
	Функциональные требования
• Зарегис	жен предоставлять возможность стрироваться. ить учетные данные пользователя.
• Просма	отривать задачи. слять свои решения в проверяющую систему.
	Требования к производительности

Часть II

Отчет

Обзор проекта

Используемые языки

Как следует из следующей инфографики, большая часть проекта была написана на языке Go. Часть на JavaScript (проверка регистрационной формы), а сами страницы оформлены с использованием HTML — верстка и CSS — разметка и дизайн.

● CSS 43.0% ● HTML 24.2% ● Go 22.4% ● JavaScript 10.4%

Рис. 1. Инфографикаа

Процентное соотношение

- CSS 43.0;
- HTML 24.2;
- Go 22.4;
- \bullet JavaScript-10.4.

------ Структура проекта ------

Связь между функциями проекта.

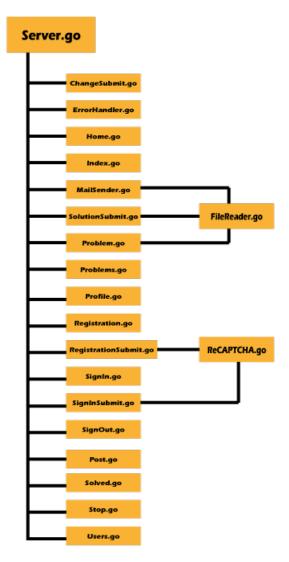


Рис. 2. Функции проекта

------ Ход работы

Пользователь заходит на стартовую страницу.



Рис. 3. Стартовая страница

Дальше он может пройти либо на страницу регистрации,

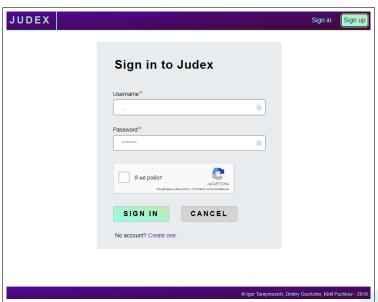


Рис. 4. Страница авторизации

либо на страницу авторизации.

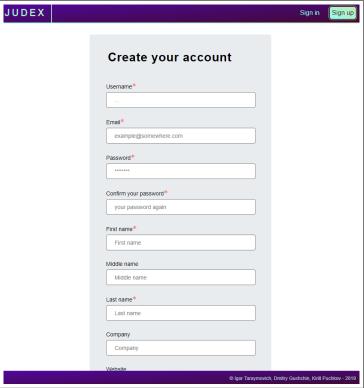


Рис. 5. Страница регистрации

После авторизации вы попадаете на эту страницу. Дальше есть возможность пройти на страницу с задачами или зайти на страницу профиля и поменять какие-нибудь данные (свои).



Рис. 6. Страница после авторизации

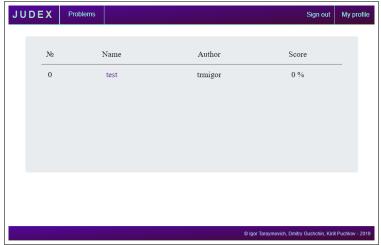


Рис. 7. Страница с задачами

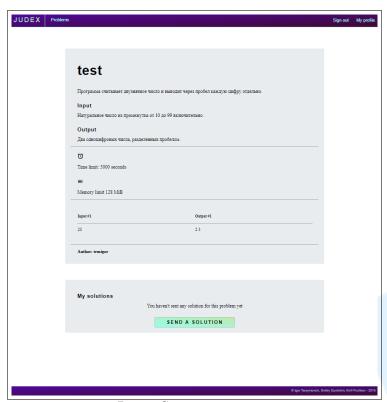


Рис. 8. Страница с задачей

Подробности реализации

..... Описание функций

5.1.1 Server.go

Любой файл в Go начинается со следующей строки.

```
1 package main
```

Дальше идет импорт нужных пакетов.

```
"fmt"
4
                              I/O\ formatting
        " i o "
5
                           // EOF
                           // Strings split
        "strings"
6
7
       "html/template"
                          // Html usage
                           // Logs
        "net"
9
                              Server logic
        "net/http"
10
                              Server logic
11
                              OS syscalls
12
       "time"
                              Timing
       "os/signal" // Signal handling
"path/filepath" // Filepath join
13
14
15
        "\,github.com/\,dpapathanasiou/go-recaptcha"
16
17
        // Database
18
        "context"
19
20
        "github.com/mongodb/mongo-go-driver/mongo"
21
```

Определяем структуру, чтобы хранить и использовать информацию, связанную с пользователем: логин, почту, пароль, имя, фамилия и прочее.

```
24 type User struct {
    Username string
```

```
Email
26
                   string
       Password
27
                   string
28
       FirstName string
29
       MiddleName string
30
       LastName
                   string
31
       Company
                   string
32
       Website
                   string
33
       Bio
                   string
34
```

Прописываем структуру для учетных данных пользователей, которые вошли в сеть.

Время, которое выделено для пользователя внутри системы, после окончания которого его выбрасывает из системы.

```
45 const CredCur time. Duration = 3 * time. Hour
```

Объявляем глобальные переменные.

```
47
48
       // Html templates storage
49
       templates \ *template. Template
50
       // MongoDB client
51
52
       client *mongo.Client
53
54
       // Server configure
55
       logPath, templatePath, staticPath, emailPatterns,
       problemsPath, solutionsPath, tmpPath string
56
57
```

Главная функция, в которой идет реализация работы сервера.