

TryMe

Онлайн подготовка на ученици

Преподавател: доц. д-р Милен Петров

Изготвили:

Емине Башева 61729

Таня Христова 61684

Христо Тодоров 61676

Моника Петева 61699

Начална година: 2016

Програма: бакалавър, (СИ)

Курс: 3

Тема: TryMe

Дата: 2016-07-01

Предмет: [wwwTech2015_16_6ed_SI_summer](#)

имейл: eeminkaa@gmail.com, tdhristova@yahoo.com, hgtodorov@gmail.com,
MonikaPetevaPetkova@gmail.com

1. Условие

Да се направи уеб сайт за подпомагане на образователния процес при обучението на деца. Сайтът трябва да предоставя възможност да могат да се регистрират лица и да имат достъп до тестове. Самите тестове трябва да могат да бъдат попълвани и в края им да се отчита резултат от решения тест. Потребителите да могат да събират точките си от решаваните тестове и да участват в класации с тях. За насърчаване на обучението да се предоставя и възможност за спечелване на значки, които да бъдат достъпвани от профила на потребителя. Профила на потребителя трябва да съдържа следната информация: име, училище, предмети, в които е най-добър, снимка и съответно баджовете, които е спечелил. Потребителя да има възможност да споделя резултатите си в края на всеки завършен тест по мейл или в социалните мрежи Facebook и Twitter.

2. Въведение

При подготовката на учениците от средното образование често се налага да се използват външни източници, като сборници с тестове, различни помагала и т.н. Всичко това е в тежест както за ученика, така и за неговите родители. Във време, в което всичко вече е дигитализирано, подготовката на учениците не би трябвало да изостава. Самите те постоянно използват интернет за всичко - слушане на музика, гледане на филми, социални мрежи. Затова по-лесно за самите тях би било ако всичките тези упражнения ги правят онлайн.

Затова и идеята за система, която да улеснява учебния процес се пренася във виртуалното пространство. Нашата идея е за система, която да

позволява на ученици да проверят своите знания по различни предмети чрез попълване на различни тестове. Доказано е, че формата тест е асоциирана с конкурентност (да имаш по-добър резултат), затова смятаме, че това би подпомогнало потребителите да учат по-добре.

Тестовите ще покриват основните предмети, считани като задължително избираеми в плановите за обучение на ученици от средното образование. Самите те ще бъдат класифицирани по класове, за по-лесно ориентиране на потребителя. Предоставя се възможност за неограничен брой тестове наименувани по един и същи начин, стига съдържанието им да е различно.

Тестовите ще могат да бъдат писани и от потребители, те обаче нямат право да ги публикуват, а само да ги изпратят на админа на сайта за одобрение. По този начин ще се следи за съдържанието на публикуваните тестове. Само потребители, ползващи се с определени привилегии ще могат да проектират и предлагат за публикация тестове.

Целта на системата е да подготви учениците по-качествено за предстоящите им писмени изпитвания, със серия от предварително подбрани въпроси. Всеки тест ще има и време за завършване, в зависимост от трудността му. Ще има и точки от попълнените тестове, като същите ще бъдат базови при определяне на новите привилегии на потребителите и при изготвянето на седмични, месечни и годишни класации.

В системата ще могат да се регистрират различни потребители. Тя ще включва потребителско име, парола и имейл.

След приключване на тест потребителите ще могат да получават резултатите и по имейл, както и да ги споделят в социалните мрежи фейсбук и туитър.

3. Използвани технологии

Използвали сме програмния език Java 8, JavaScript, HTML, CSS, Bootstrap 3 и MongoDB. По време на процеса на разработка сме използвали и Maven, Eclipse IDE, Tomcat 8.

4. Инсталация и настройки

4.1. Предварителни изисквания

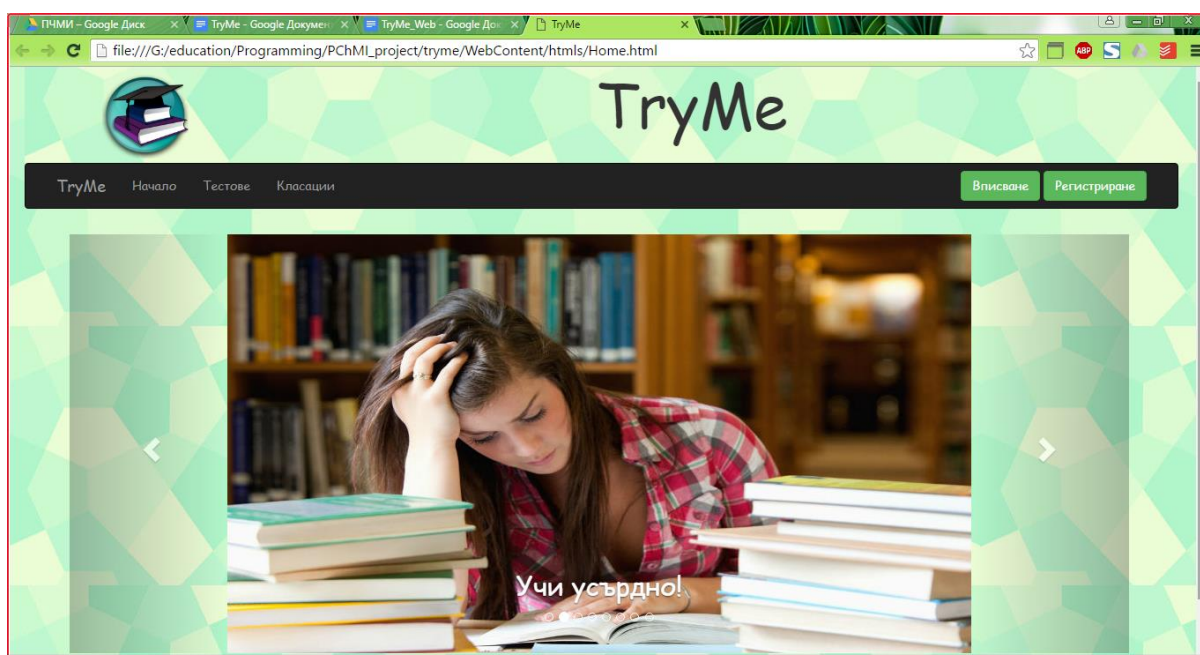
- Java SDK 1.8.xx (Уверете се, че имате системна променлива JAVA_HOME, която сочи към /bin директорията на SDK-то)
- Maven 3.3.9 (Уверете се, че имате добавена към системната променлива Path пътят до maven).
- Apache Tomcat 8
- MongoDB (за инсталация и настройки се обърнете към следния линк <https://github.com/pchmi-t/tryme/wiki/How-to-install-and-configure-a-MongoDB>)

1. Свалете source кода от <https://github.com/pchmi-t/tryme/archive/1.0.0.zip> .

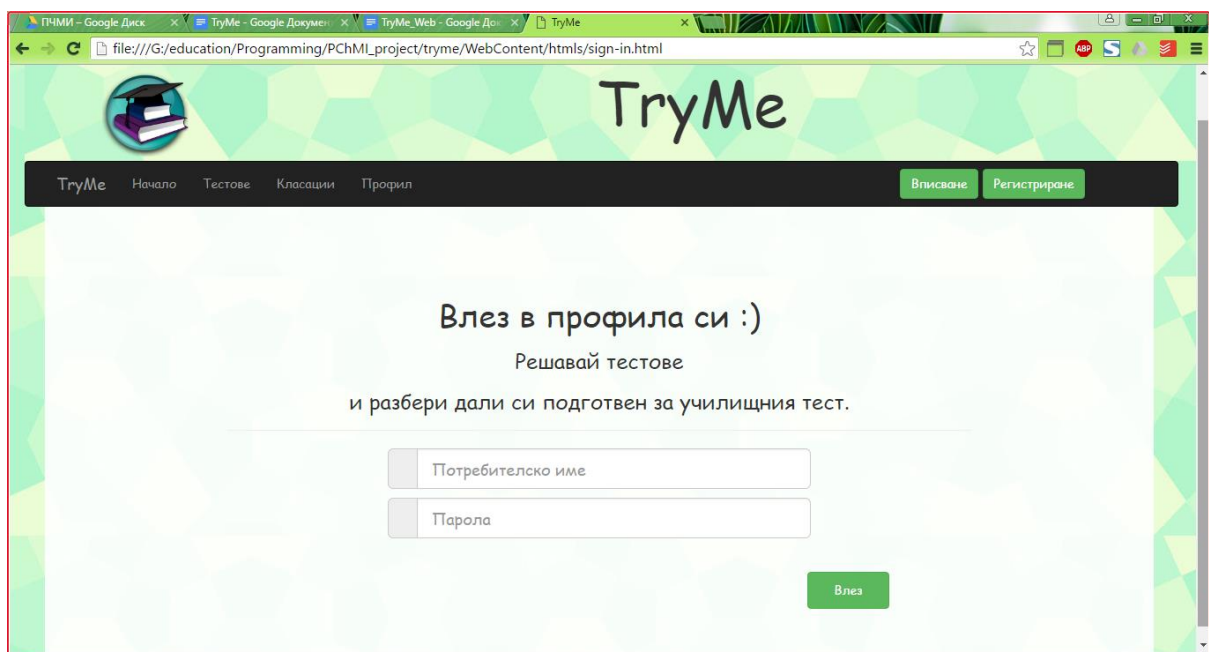
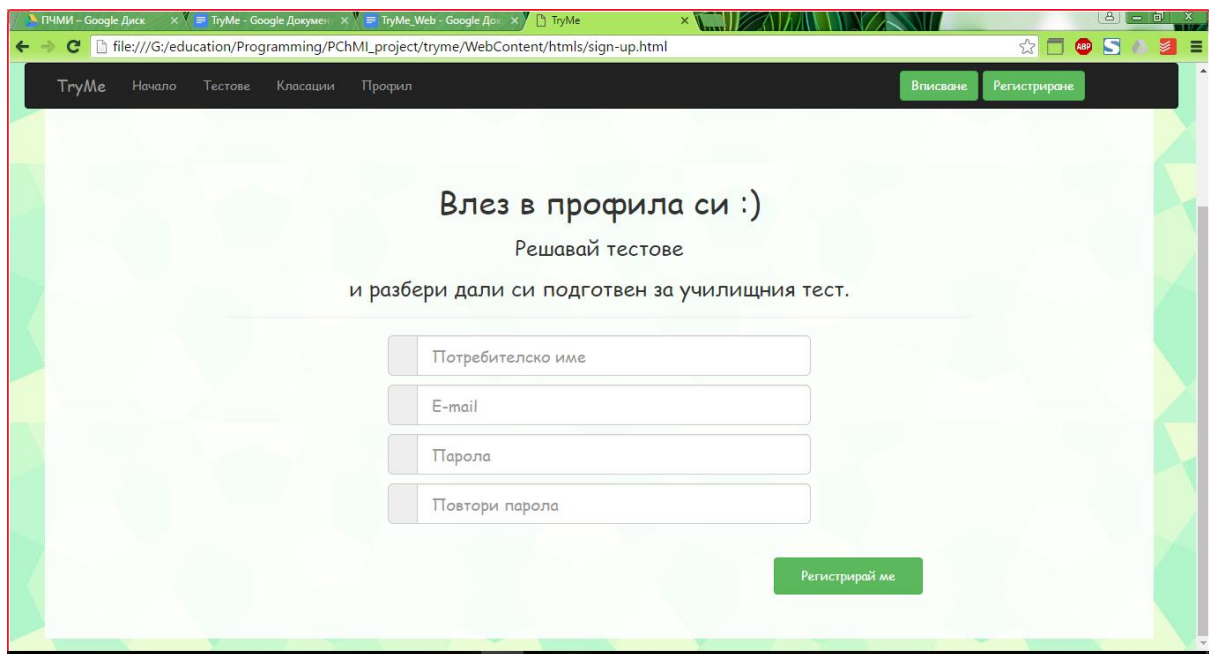
2. Отворете Eclipse и импортнете проекта като zip.
3. Навигирайте до папката на проекта в Eclipse.
4. Напишете следната команда *mvn clean install* (няма засега тестове, но за всеки случай изпълнете командата с параметри *-DskipTests*)
5. Навигирайте до Run on Server и сетъпнете сървър
6. Натиснете Start server.
7. Отворете localhost:8080/tryme/htmls/Home.html

5. Кратко ръководство на потребителя

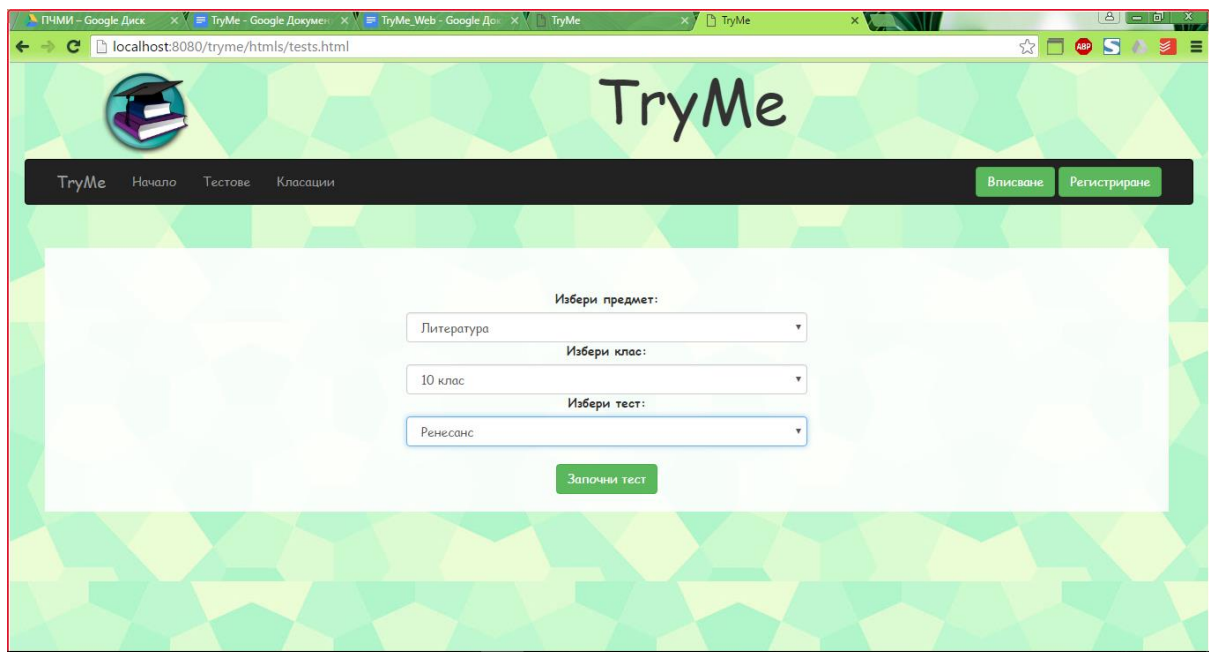
Сайта може да се достъпи на WebContent/htmls/Home.html когато не се пуска от сървър, а при наличието на такъв линка, който отговаря на съдържанието му е <http://localhost:8080/tryme/htmls/Home.html> . Началната страница изглежда така:



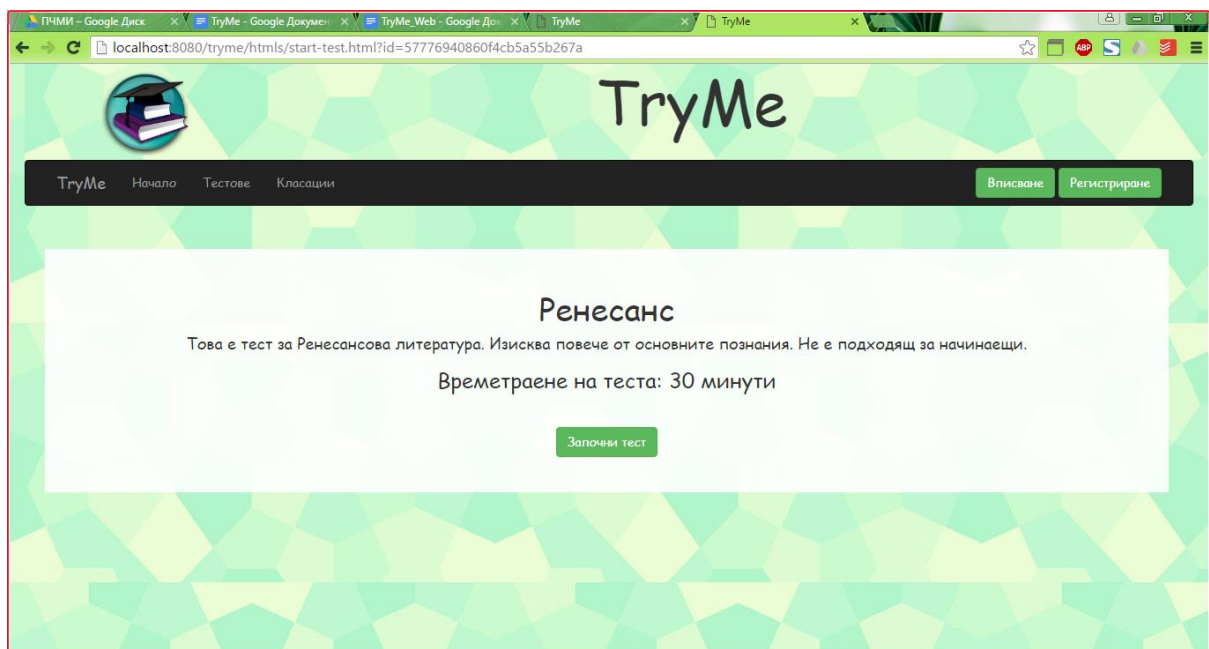
След като вече е заредена началната страница на проекта би било желателно да се влезе с някакъв акаунт, за да може да се използва пълната функционалност на сайта. За тази цел сайта предоставя регистрация и влизане чрез вече регистриран профил. Съответно страниците изглеждат така:



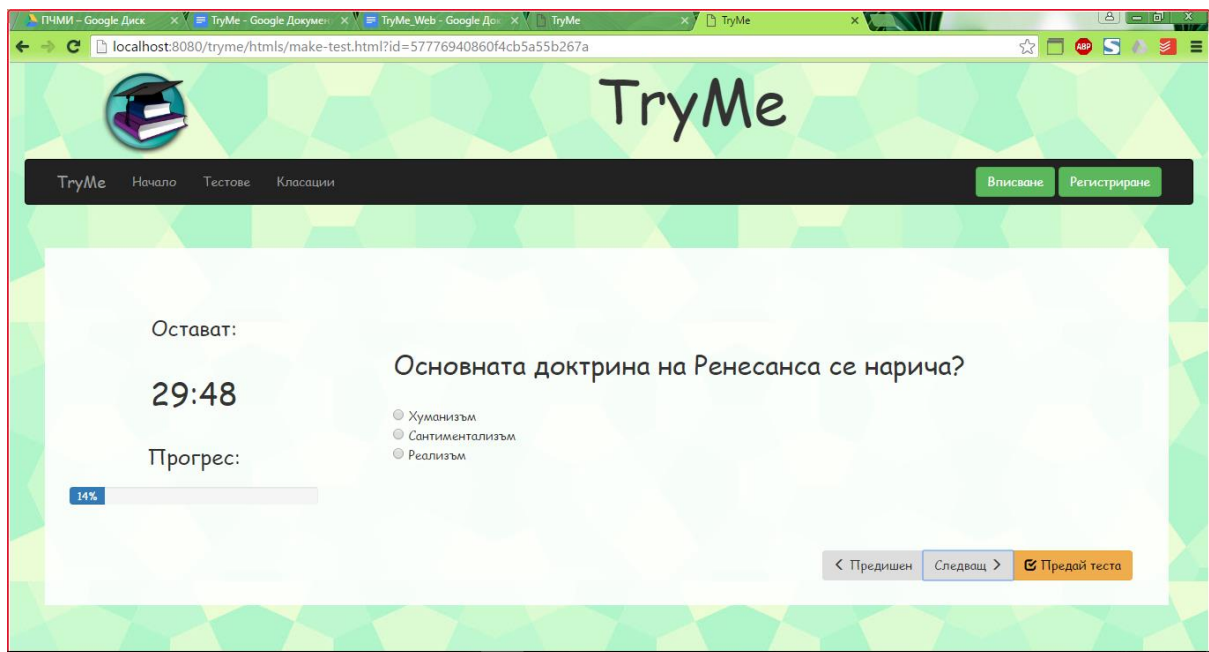
След като сте се аутентикирали пред системата имате възможност да се решават тестове. Тях може да ги намерите на <http://localhost:8080/tryme/htmls/tests.html> . Тук потребителя последователно избира предмет и клас и спрямо тях се генерират темите от избраните критерии. Ако теми няма се показва съобщение, че няма. Страницата за избор на тест изглежда така:



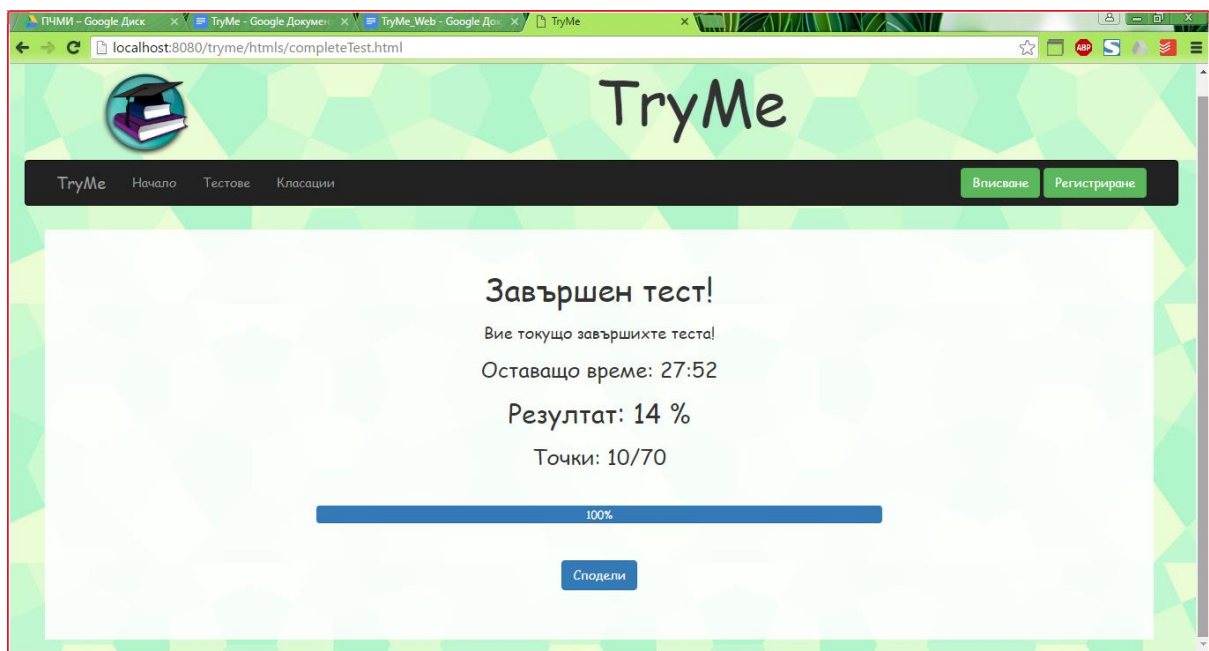
След това се потребителя се препраща към страница за започване на тест, където има информация за теста



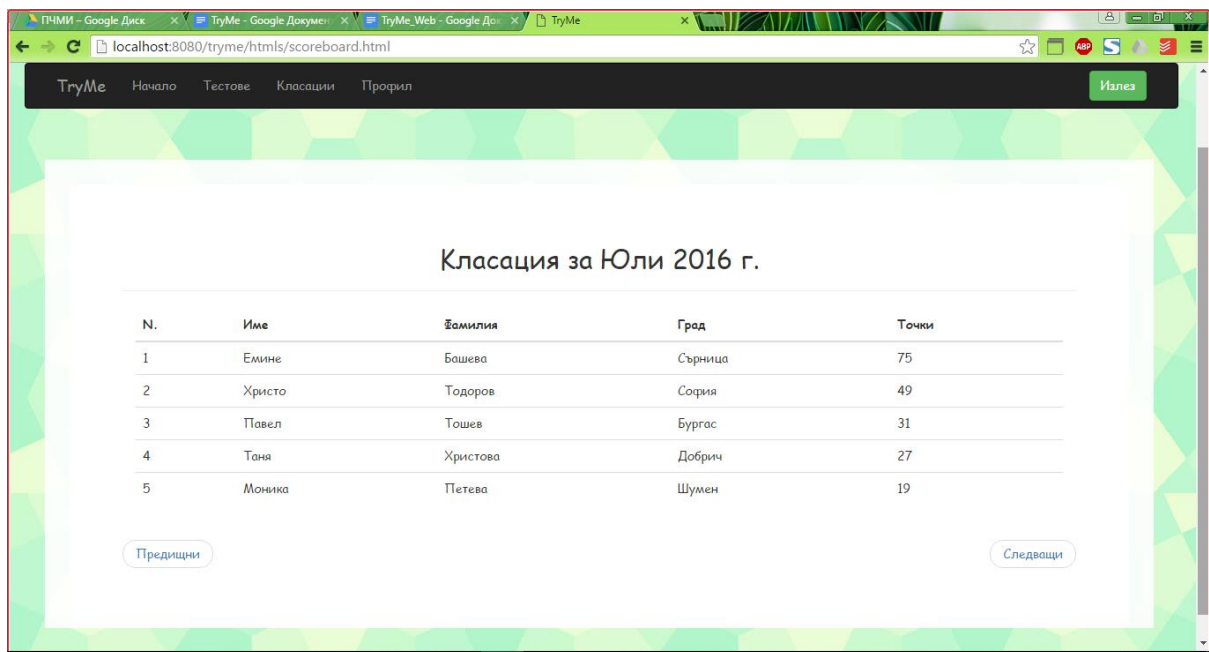
След това потребителя започва своя тест, като му се показва колко време му остава до края на теста и колко от теста е попълнил, така той да може да си управлява времето за теста и да не губи твърде много време.



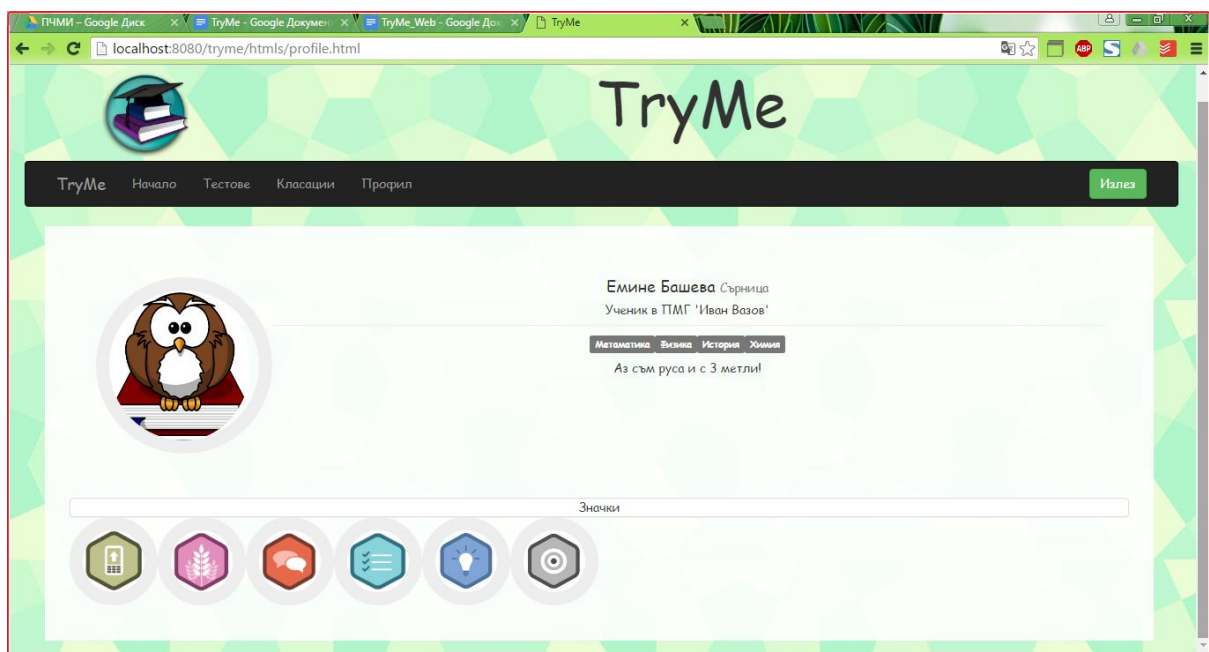
След като свършат въпросите потребителя трябва да приключи теста си с бутона “Предай тест”. Тогава сайта го отвежда на страница, която му казва колко време му е оставало до завършването на тест, какъв резултат има и колко точки е спечелил. Тук потребителя ще има възможност и да сподели резултата си.



При решаването на тест се увеличават точките на потребителя и така той участва в класация. Класацията може да се намери на адрес : <http://localhost:8080/tryme/htmls/scoreboard.html> .



Профила на потребителя може да се достъпи на <http://localhost:8080/tryme/htmls/profile.html>.



6. Описание на програмния код

В проекта файловете за UI частта се намират в папката WebContent. Там се намират html, js, css и img файловете, които се използват при зареждането на сайта. Всяка страница си има собствен html и ако е нужно css файл, който да го настройва. За управление на статичните контейнери в html страниците са създадени js файлове, които с помощта

на jQuery добавят, махат или управляват вече съществуващата информация. Всеки html файл има задължително navbar и div, който е контейнер на страницата, като той е различен за всяка такава.

Backend

Използва се абстракция над базата, под формата на Managers класове.

```
package com.tryme.managers;
```

```
import java.util.List;
```

```
import org.springframework.data.mongodb.core.query.Criteria;
```

```
import org.springframework.data.mongodb.core.query.Query;
```

```
import com.tryme.core.exceptions.InvalidAccountException;
```

```
import com.tryme.core.exceptions.NoSuchAccountException;
```

```
import com.tryme.framework.bean.Account;
```

```
import com.tryme.framework.bean.UserInformation;
```

```
import com.tryme.framework.criteria.AccountCriterion;
```

```
/**
```

```
 * Account manager for managing the operations related to the account entity
```

```
 *
```

```
 */
```

```
public interface AccountManager {
```

```
    /**
```

```
     * Update the existing account.
```

```
     *
```

```
     * @param account the account that is about to be update
```

```
     * @return the updated account
```

```
     * @throws NoSuchAccountException in case there is no such account
```

```
     * @throws IllegalIDFormatException in case the id is invalid by some
```

```
reasons
```

```
     */
```

```
    void updateAccount(Account account) throws Exception;
```

```
    /**
```

```
     * Adding an account.
```

```
     *
```

```
     * @param account the account to be added
```

```
     * @return the added account
```

```
     * @throws InvalidAccountException in case the account is not valid
```

```
     * @throws Exception
```

```
     * @throws DuplicateAccountException in case the user already exist
```

```
     */
```

```

void addAccount(Account account) throws Exception;

/**
 * Get all the accounts specified by the account criteria.
 *
 * @param criterion the account criteria
 * @return the all the accounts matched the given criteria
 * @throws Exception
 */
List<Account> getAccounts(AccountCriterion criterion, int limit) throws
Exception;

/**
 * Get the account criteria
 *
 * @return the account criteria to search by
 */
AccountCriterion getAccountCriterion();

/**
 * Get the account represented by his id.
 *
 * @param id the account's id
 * @return the account with the given id.
 */
Account getAccount(AccountCriterion criterion) throws Exception;

}
Друга част от тази абстракция за т.нар Criterion
package com.tryme.framework.criteria;

```

```

import org.bson.types.ObjectId;
import org.springframework.data.mongodb.core.query.Query;

/**
 * An account criteria to search for an accounts.
 *
 */
public interface AccountCriterion extends BaseCriteria {

/**
 * Add account 'id' criteria to search.
 *
 * @param objectId the account id

```

```

        * @return the account criteria
        */
        AccountCriterion id(String objectId);

        /**
         * Add account 'username' criteria to search.
         *
         * @param username the account username
         * @return the account criteria
         */
        AccountCriterion username(String username);

        /**
         * Add account 'email' criteria to search.
         *
         * @param email the account email
         * @return the account criteria
         */
        AccountCriterion email(String email);

        /**
         * Add the password criteria to search by.
         *
         * @param password the account's password in plain text
         * @return the account criteria
         */
        AccountCriterion password(String password);
    }

```

Следва и тяхната имплементация. Използва се Factory имплементация, за да се вземат техните имплементации. Използва се REST ресурси, за да може да се правят извиквания от страна на фронтенда. Връзката и транзакциите с базата се осъществяват с помощта на Session класа:

```

package com.tryme.core;

import java.net.UnknownHostException;

import org.springframework.data.mongodb.core.MongoOperations;
import org.springframework.data.mongodb.core.MongoTemplate;

import com.mongodb.MongoClient;
import com.tryme.constants.CoreConstants;

```

```

public class Session implements AutoCloseable {

    private MongoOperations session;

    private MongoClient connection;

    /**
     * Open the new db session.
     *
     * @return the opened db session
     * @throws UnknownHostException in case the unknown or
     unreachable host is specified
     */
    public MongoOperations openSession() throws
    UnknownHostException {
        this.connection = new MongoClient(CoreConstants.DB_HOST,
CoreConstants.DB_PORT);
        this.session = new MongoTemplate(connection,
CoreConstants.TRYME_DB_NAME);
        return this.session;
    }

    /**
     * Close the opened session.
     *
     */
    public void closeSession() {
        if(connection != null) {
            connection.close();
        }
    }

    @Override
    public void close() throws Exception {
        closeSession();
    }

}

```

7. Какво научихме

Най-много научихме за JavaScript и за основната му библиотека jQuery без, която създаването на уеб страница се свежда до жалки опити за управление на който и да е елемент. За някои от нас (незнаещи що е да пишеш запитвания към сървъра) беше полезно да научим и ајах заявки, които да достъпват ресурсите, запазени в нашата база от данни. За база от данни използвахме MongoDB. Избрахме я, защото не бяхме работили с NoSQL база данни и по този начин щеше да имаме възможност да придобием знания и за това.

11. Използвани източници

- <http://www.w3schools.com/jquery/default.asp> - това е курса за jQuery в w3school, който ние изкарахме, за да можем да пишем релевантен код
- <http://www.w3schools.com/js/> - както и курса по JavaScript, за да освежим знанията си и да попълним пропуските, които имаме
- http://www.tutorialspoint.com/java_technology_tutorials.htm - кратки уроци по основните JAVA технологии

Предал:

/фн, имена/

Приел:

/доц. Милен Петров/