Кыргызский Государственный Технический Университет

имени И.Разакова

Факультет информационных Технологий

Программная Инженерия

Средство Визуальной

Разработки ПО

выполнил: Абакиров Нурсултан

группа: Пи(а) 1-16

**Генератор Билетов**

**Задача**

Создать сайт, который поможет генерировать экзаменационные билеты. Сохранять введенные предметы и вопросы в личном кабинете. Генерировать билеты по введенным параметрам: количество билетов, количество вопросов в каждом билете. Выводить сгенерированные билеты в формате Office Word

**Требования**

* мультиюзерность - логин, регистрация
* сохранять данные (пользователь, предметы пользователя, вопросы пользователя)
* выдавать данные по запросам

**Реализация**

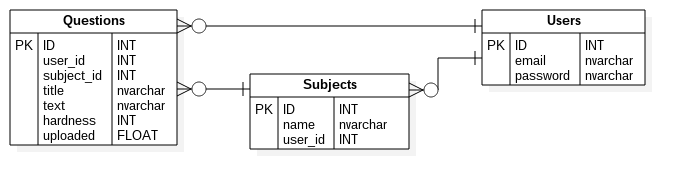
***Backend (серверная часть)***

* Язык программирования - Python3.5
* web framework - Flask
* database - sqlite3

***Frontend (веб сайт)***

* html
* js
* css
* axios framework
* materializecss framework

**База Данных**



*Поток данных реализован по стандарту API, благодаря этому можно не ограничиваться одним веб-сайтом, можно добавить различные клиенты (windows, macos, android, ios)*

**API**

**Registration**

*/api/register*

request

POST:

headers:

{

“Content-Type”: ”application/json”

}

Payload:

{

“email”: “email”,

“password”: “password”

}

response

{

"data": {

"data": {

"email": "email",

"id": user id

}

},

"message": "OK",

"status": 200,

"token": "access\_token"

}

**Login**

*/api/login*

request

POST:

headers:

{

“Content-Type”: ”application/json”

}

Payload:

{

“email”: “email”,

“password”: “password”

}

response

{

"data": {

"email": "email",

"id": user\_id

},

"message": "OK",

"status": 200,

"token": "access\_token"

}

**Token verification**

*/api/verify*

request

POST:

headers:

{

“Content-Type”: ”application/json”

}

Payload:

{

“token”: “access\_token”

}

GET:

params:

token = access\_token

response

{

"data": {

"email": "email",

"id": user\_id

},

"message": "OK",

"status": 200

}

**get list of subjects**

*/api/<user\_id>*

request

GET:

params:

token = access\_token

response

{

"data": [

{

"id": subject\_id,

"name": "subject\_name"

},

{

"id": subject\_id,

"name": "subject\_name"

},

...

],

"message": "OK",

"status": 200

}

**save new subject**

*/api/<user\_id>*

request

POST:

headers:

{

“Authorization”: “Bearer access\_token”

}

payload:

{

"name": "subject\_name",

"user\_id": user\_id

}

response

{

"message": "OK",

"status": 200

}

**update existing subject**

*/api/<user\_id>*

request

POST:

headers:

{

“Authorization”: “Bearer access\_token”

}

payload:

{

“id”: subject\_id

"name": "subject\_name",

"user\_id": user\_id

}

response

{

"message": "OK",

"status": 200

}

**get list of questions**

*/api/<user\_id>/<subject\_id>*

request

GET:

params:

token = access\_token

response

{

"data": [

{

"hardness": hardness\_level,

"id": question\_id,

"subject\_id": subject\_id,

"text": "question text",

"uploaded": uploaded time,

"user\_id": user\_id

},

{

"hardness": hardness\_level,

"id": question\_id,

"subject\_id": subject\_id,

"text": "question text",

"uploaded": uploaded time,

"user\_id": user\_id

},

...

],

"message": "OK",

"status": 200

}

**save new question**

*/api/<user\_id>/<subject\_id>*

request

POST:

headers:

{

“Authorization”: “Bearer access\_token”

}

payload:

{

"text": "question text",

"hardness": hardness level

}

response

{

"message": "OK",

"status": 200

}

**update existing question**

*/api/<user\_id>/<subject\_id>*

request

POST:

headers:

{

“Authorization”: “Bearer access\_token”

}

payload:

{

“id”: question\_id

"text": "question text",

"hardness": hardness level

}

response

{

"message": "OK",

"status": 200

}

**generate tickets**

*/api/generate/<user\_id>/<subject\_id>*

request

POST:

headers:

{

“Authorization”: “Bearer access\_token”

}

payload:

{

"ticket\_cnt": count of tickets,

"question\_cnt": count of questions in each ticket

}

response

{

"data": {

"q\_count": count of questions in each ticket,

"subject\_id": "subject\_id",

"t\_count": count of tickets,

"tickets": [

{

"questions": [

{

"hardness": hardness level,

"q\_number": question number,

"text": "question text"

},

{

"hardness": hardness level,

"q\_number": question number,

"text": "question text"

}

],

"t\_number": ticket number

},

{

"questions": [

{

"hardness": hardness level,

"q\_number": question number,

"text": "question text"

},

{

"hardness": hardness level,

"q\_number": question number,

"text": "question text"

}

],

"t\_number": ticket number

}

],

"user\_id": "user\_id"

},

"download": "download link",

"filename": "filename",

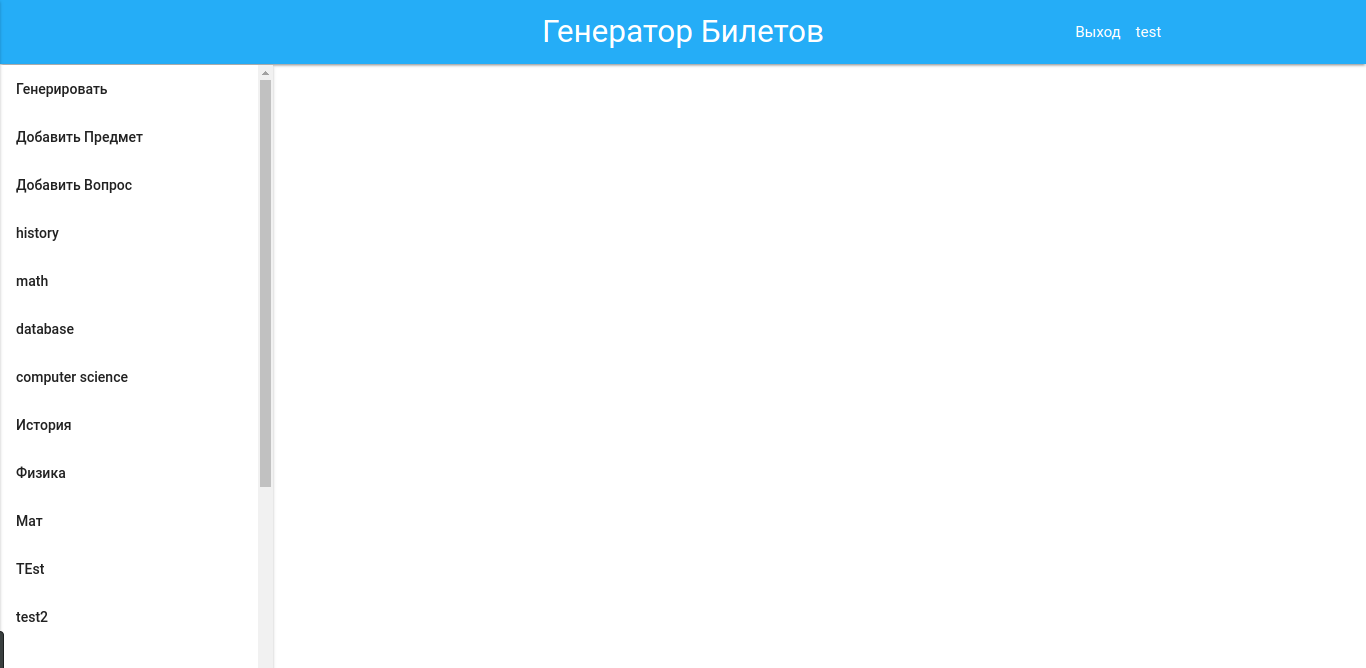
"message": "OK",

"status": 200

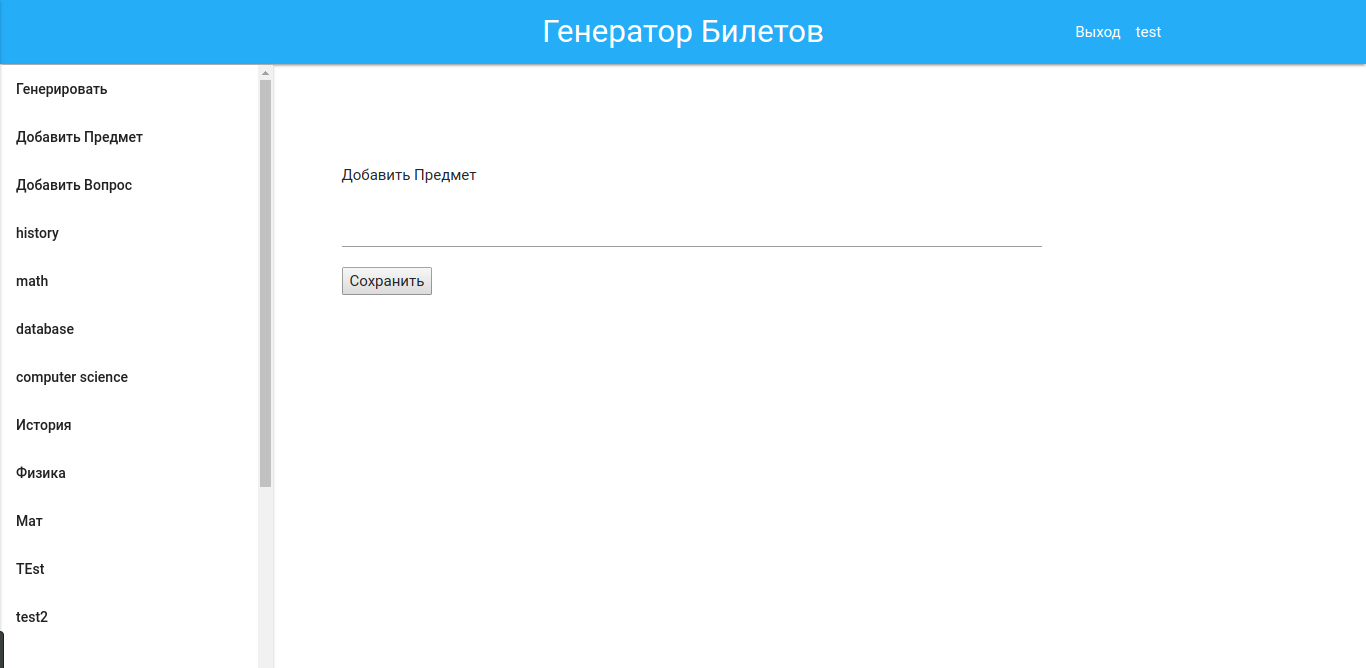
}

**UI**

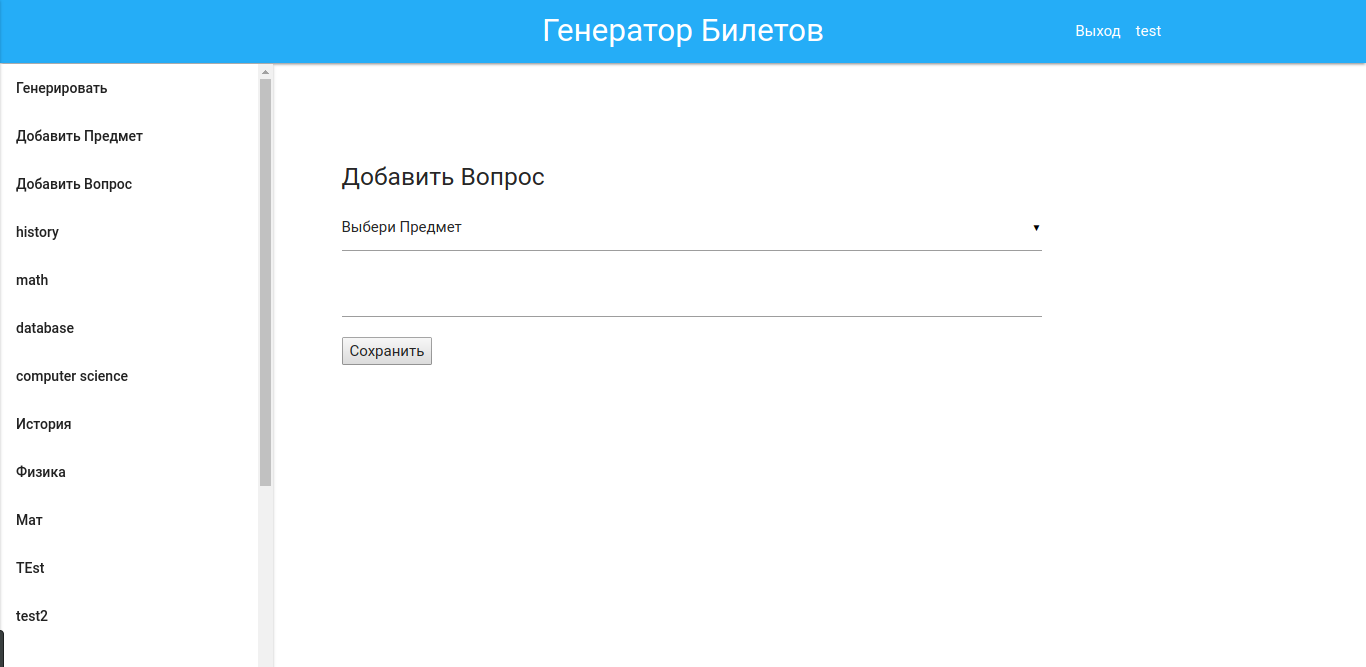
*главная страница*



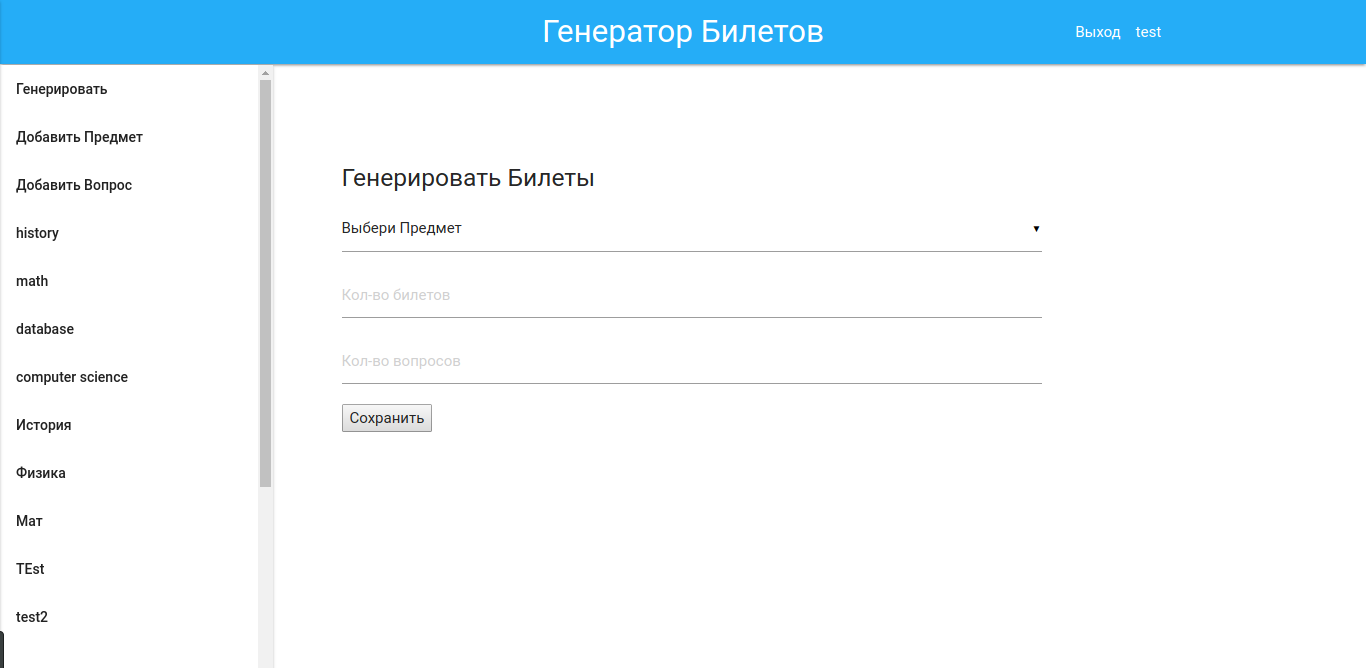
*страница добавления предмета*

**

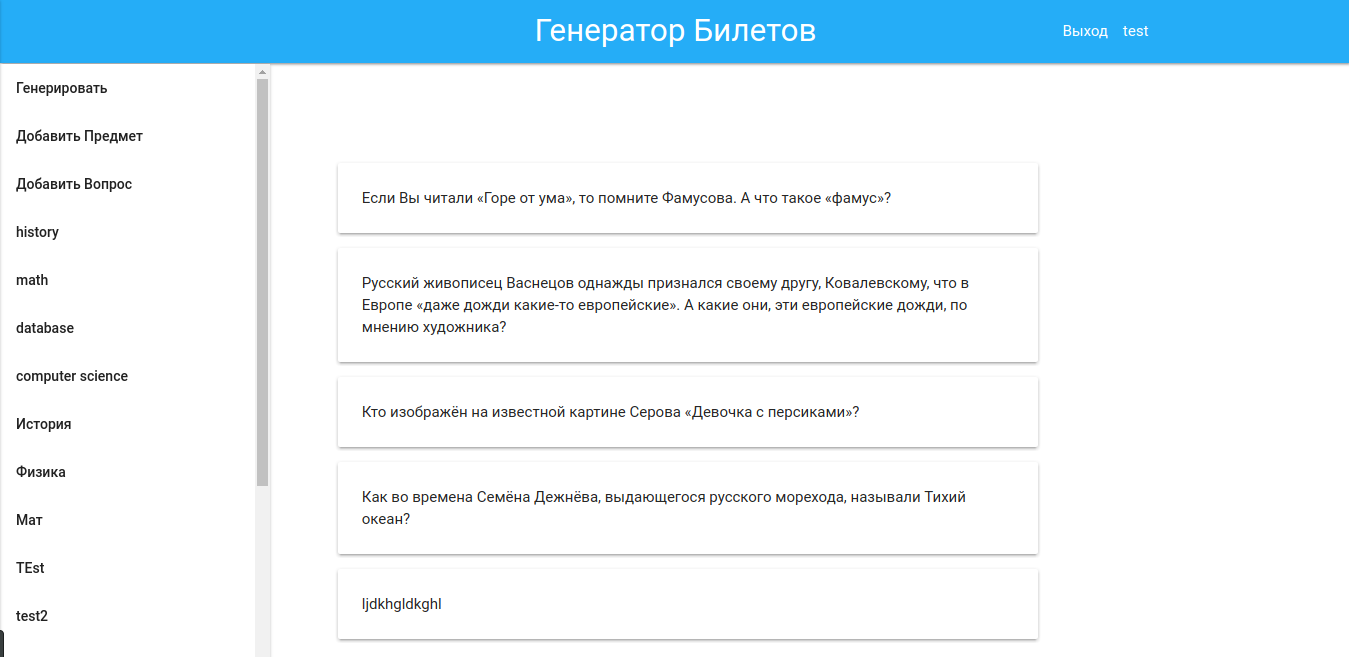
*страница добавления вопроса*

**

*страница генерации*

**

*список вопросов*

**

***Пример сгенерированного файла***

# ***Билет № 1***

*1. ljdkhgldkghl*

*2. Если Вы читали «Горе от ума», то помните Фамусова. А что такое «фамус»?*

*3. Кто изображён на известной картине Серова «Девочка с персиками»?*

*4. Как во времена Семёна Дежнёва, выдающегося русского морехода, называли Тихий океан?*

*5. Русский живописец Васнецов однажды признался своему другу, Ковалевскому, что в Европе «даже дожди какие-то европейские». А какие они, эти европейские дожди, по мнению художника?*

# ***Билет № 2***

*1. ljdkhgldkghl*

*2. Как во времена Семёна Дежнёва, выдающегося русского морехода, называли Тихий океан?*

*3. Кто изображён на известной картине Серова «Девочка с персиками»?*

*4. Русский живописец Васнецов однажды признался своему другу, Ковалевскому, что в Европе «даже дожди какие-то европейские». А какие они, эти европейские дожди, по мнению художника?*

*5. Если Вы читали «Горе от ума», то помните Фамусова. А что такое «фамус»?*