МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РФ

Федеральное государственное автономное образовательное
учреждение высшего образования
ЮЖНЫЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ

Институт математики, механики и компьютерных наук
имени И. И. Воровича

Направление подготовки
02.03.02 — Фундаментальная информатика
и информационные технологии

РАЗРАБОТКА СИСТЕМЫ ПРОВЕДЕНИЯ ОПРОСОВ АУДИТОРИИ ВО ВРЕМЯ ПУБЛИЧНЫХ ВЫСТУПЛЕНИЙ

Выпускная квалификационная работа
на степень бакалавра

Студента 4 курса
Е. А. Тактарова

Научный руководитель:
к.ф.-м.н., доцент Е. М. Андреева

Допущено к защите:
руководитель направления ФИИТ _____ В. С. Пилиди

Ростов-на-Дону
2019

Содержание

Введение

Технологии проведения публичных выступлений и презентаций затрагивают навыки ораторства и внешний вид, дизайн медиасопровождения. Методы взаимодействия с аудиторией традиционно включают в первую составляющую. Выступающий, желающий взаимодействовать со слушающими, должен уже обладать определенным опытом в работе с ними и ограничен устными средствами.Крайне редко возможно почти полностью вовлечь аудиторию в выступление, ведь лишь немногие слушатели готовы,например, задать вопрос или ответить выступающему.

Распространение телефонов и мобильного доступа в интернет, позволяет использовать эти устройства как средства взаимодействия с аудиторией. Проекты, использующие эту идею, реализовывались неоднократно, но ни один из них не закрепился как широко используемый в публичных выступлениях. В первую очередь, идея взаимодействия с публикой через телефоны реализовывалась под конкретные единичные выступления. Последующие реализации, хотя и обладают обширным функционалом, в виде опросов, голосований и чатов, представлют собой отдельные веб-сервисы, направленные на монетизацию с пользователей. Все проекты закрыты проприетарными лицензиями и требуют от пользователей загрузки презентации на сторонний сервер.

Данная работа посвящена разработке проекта портативного веб-сервиса под свободной лицензией, который позволит проводить опросы аудитории во время публичных выступлений без привлечения сторонних сервисов. Свободная лицензия позволит любому человеку изменять и расширять возможности сервиса под свои нужды.

Задача по созданию такого проекта включает как разработку веб-интерфейса пользователя (фронтенд), так и разработку внутренней логики сервиса(бэкенд), которые в совокупности обеспечат динамичное отображение результатов опросов.

1. Исследование предметной области

1.1. Обзор существующих решений

Как и упоминалось ранее, для опросов аудитории уже существует немалое число инструментов, однако в основой массе это закрытые решения в виде веб-сервисов:

- polleverywhere.com
- directpoll.com
- sli.do
- ficus.io

На этих сайтах и других подобных можно бесплатно в первый раз провести опрос или даже презентацию, но повторные показы и дополнительные функции ограничены для пользователей, не оплативших услуги сайтов. Более того, даже оплативший пользователь ограничен средствами и функциями сайта и не может модифицировать или изменить инструмент под свои нужды и цели.

Также стоит упомянуть об инструментах опросов, не использующих только Интернет(http://www.ombea.com/). Такие решения применяются в университетах США(http://www.nea.org/home/34690.html) и отличаются низкой способностью к масштабированию и высокой ценной как системы, так и индивидуальных приборов голосования.

1.2. Обзор инструментов разработки

При создании веб-сервиса самую важную роль занимает разработка серверной части. Так как веб-сайт должен динамически взаимодействовать с сервером, то архаичная связка из веб-сервера и FastCGI/CGI приложения очевидно не подойдет. Для решения данной задачи необходимо выбрать один из множества современных веб-

фреймворков (https://ru.wikipedia.org/wiki/Сравнение_каркасов_веб-приложений),как основу для проекта. Отметим основные необходимые для задачи черты фреймворков:

- легковесность
- инкапсуляция веб-сервера
- наличие актуального функционала(JSON,AJAX,websocket)

Рассмотрим несколько популярных фреймворков:

- Django — фреймворк на языке Python. Хотя на нем можно реализовать необходимый нам функционал, но его врядли можно назвать легковесным. Django в первую очередь предназначен для создания больших многостраничных сайтов и сервисов, которые будет длительное время поддерживать команда разработчиков и администраторов. Наличие бесполезного для задачи функционалла негативно сказывается на времени освоения и разработки. (https://www.djangoproject.com/)
- Ruby on Rails — фреймворк на языке Ruby. Основными минусами Ruby on Rails являются проксирование через отдельный веб-сервер и общая сложность освоения как фреймворка, так и самого языка.Стоит также отметить, что этот фреймворк сильно опирается на архитектуру модель-представление-контроллер, реализация которой усложняет задачу для небольшого приложения. (https://rubyonrails.org/)
- Express — фреймворк на языке Javasciprt, запускаемый на платоформе Node.js. Express инкапсулирует веб-сервер,представляя только абстракцию в виде объектов HTTP запроса и ответа,а необходимый для приложения функционал, например WebSocket и шаблонизация , добавляются через совместимы модули Node.js. Так же фреймворк не затрагивает клиентскую часть веб-приложения.(https://expressjs.com/ru/) Express

своевременно обновляется, имеет обширную документацию и является популярным выбором среди разработчиков из-за своей простоты и понятности. Основным опасением является производительность однопоточной архитектуры Node.js, однако для небольших и средних приложений Node.js и Express показываю удовлетворительные результаты(https://www.researchgate.net/publication/286594024_Performance_Compa

2. Обзор Проекта

3. Несколько примеров в LaTeX

Некоторые часто используемые команды приведены в качестве примера ниже (и варианты — в комментариях). Мы рекомендуем внимательно прочесть данный текст и изучить его исходный код прежде, чем начинать писать свой собственный. Кроме того, можно дать и такой совет: идущий ниже текст не убирать до самого конца, а просто оставлять его позади своего собственного текста, чтобы в любой момент можно было проконсультироваться с данными примерами.

3.1. Как вставлять листинги и рисунки

Для крупных листингов есть два способа. Первый красивый, но в нём не допускается кириллица (у вас может встречаться в комментариях и печатаемых сообщениях), он представлен на листинге 3.1.

Листинг 3.1. Программа “Hello, world” на C++

```
#include <iostream>
using namespace std;

int main()
{
    cout << "Hello, world" << endl;
    system("pause");
    return 0;
}
```

Второй не такой красивый, но без ограничений (см. листинг 3.2).

Листинг 3.2. Программа “Hello, world” без подсветки

```
#include <iostream>
using namespace std;

int main()
{
    cout << "Привет, мир" << endl;
}
```

Можно использовать первый для вставки небольших фрагментов внутри текста, а второй для вставки полного кода в приложении, если таковое имеется.

Если нужно вставить совсем короткий пример кода (одна или две строки), то выделение линейками и нумерация может смотреться чересчур громоздко. В таких случаях можно использовать окружения `lstlisting` или `Verb` без `ListingEnv`. Приведём такой пример с указанием языка программирования, отличного от заданного по умолчанию:

```
fibs = 0 : 1 : zipWith (+) fibs (tail fibs)
```

Таблица 1 — Подпись к таблице — сверху

Item		
Животное	Описание	Цена ($)
Gnat	per gram	13.65
	each	0.01
Gnu	stuffed	92.50
Emu	stuffed	33.33
Armadillo	frozen	8.99

Такое решение — со вставкой нумерованных листингов покрупнее и вставок без выделения для маленьких фрагментов — выбрано, например, в книге Эндрю Таненбаума и Тодда Остина по архитектуре компьютера [1] (см. рис. 1).

Наконец, для оформления идентификаторов внутри строк (функция `main` и тому подобное) используется `lstinline` или, самое простое, моноширинный текст (`\texttt`).

Использовать внешние файлы (например, рисунки) можно и на overleaf.com: ищите кнопочку upload.

3.2. Как оформить таблицу

Для таблиц обычно используются окружения table и tabular — см. таблицу 1. Внутри окружения tabular используются специальные команды пакета booktabs — они очень красивые; самое главное: использование вертикальных линеек считается моветоном.

3.3. Как набирать формулы

LaTeX is great at typesetting mathematics. Let $X_1, X_2, \ldots, X_n$ be a sequence of independent and identically distributed random variables

После вызова этой процедуры решение должно выводиться на экран. Сначала процедура проверяет, равно ли единице значение n. Если да, то решение тривиально: нужно просто переместить один диск с i на j. Если n не равно 1, решение состоит из трех частей и каждая из этих частей представляет собой рекурсивную процедуру.

Все решение представлено в листинге 5.6. Рассмотрим такой вызов процедуры:

```
towers (3, 1, 3)
```

Этот вызов порождает еще три вызова:

```
towers (2, 1, 2)
towers (1, 1, 3)
towers (2, 2, 3)
```

Первый и третий вызов производят по три вызова каждый, и всего получится семь.

Листинг 5.6. Процедура для решения задачи «Ханойская башня»

```
public void towers (int n, int i, int j) {
  int k;
  if (n == 1)
    System.out.println("Переместить диск с " + i + "на" + j);
  else {
    k=6-i-j;
    towers(n-1, i, k);
    towers (1, i, j);
    towers (n-1, k, j);
  }
}
```

Для рекурсивных процедур нам нужен стек, чтобы, как и в IJVM, хранить параметры и локальные переменные каждого вызова. Каждый раз при вызове процедуры на вершине стека располагается новый стековый кадр для процедуры. Текущий кадр — это кадр, созданный последним. В наших примерах стек растет

Рисунок 1 — Пример оформления листингов в [1]

with $\mathrm{E}[X_i] = \mu$ and $\mathrm{Var}[X_i] = \sigma^2 < \infty$, and let

$$S_n = \frac{X_1 + X_2 + \cdots + X_n}{n} = \frac{1}{n}\sum_{i}^{n} X_i$$

denote their mean. Then as n approaches infinity, the random variables $\sqrt{n}(S_n - \mu)$ converge in distribution to a normal $\mathcal{N}(0, \sigma^2)$.

3.4. Как оформлять списки

Нумерованные списки (окружение enumerate, команды item)…

1. Like this,
2. and like this.

…маркированные списки …

- Like this,
- and like this.

…списки-описания …

Word Definition

Concept Explanation

Idea Text

Заключение

Помните, что на все пункты списка литературы должны быть ссылки. LaTeX просто не добавит информацию об издании из bib-файла, если на это издание нет ссылки в тексте. Часто студенты используют в работе электронные ресурсы: в этом нет ничего зазорного при

одном условии: при каждом заимствовании следует ставить соответствующую ссылку. В качестве примера приведём ссылку на сайт нашего института [2].

Для дальнейшего изучения LaTeX рекомендуем книгу Львовского [3]: она хорошо написана, хотя и несколько устарела. Обычно стоит искать подсказки на tex.stackexchange.com, а также читать документацию по установленным пакетам с помощью команды

```
texdoc имя_пакета
```

или на ctan.org.

Список литературы

1. *Таненбаум Э.*, *Остин Т.* Архитектура компьютера. — 6-е изд. — СПб. : Питер, 2013.
2. Сайт Мехмата ЮФУ. — URL: `http://mmcs.sfedu.ru` (дата обр. 01.01.2015).
3. *Львовский С. М.* Набор и вёрстка в системе LaTeX. — М. : МЦНМО, 2006. — URL: `http://www.mccme.ru/free-books/llang/newllang.pdf`.

Приложение А. Пример работы программы

Здесь длинный листинг с примером работы.