

Содержание

Аннотация	3
1 Введение	4
1.1 Актуальность и значимость проблемы	4
1.2 Описание предметной области	4
1.2.1 L ^A T _E X	4
1.2.2 Telegram-бот	4
1.3 Постановка задачи	4
2 Обзор литературы	6
3 Работа с Telegram-ботом	7
4 Работа с библиотекой PyLaTeX	9
5 Работа программы	10
5.1 Команды	10
5.2 Ход работы	11
5.3 Компиляция PDF-файла	11
6 Заключение	13
7 Приложения	13
Список литературы	14

Аннотация

В данной работе я реализовал Telegram-бота, который помогает студентам 2 курса ПМИ написать итоговый отчет по курсовому проекту, позволяя им не беспокоясь о конечном визуальном облике документа, полностью сосредоточиться на содержании и просто и быстро изложить результат своей работы.

С помощью подсказок бота пользователь вводит информацию, пока её не хватит для завершения работы. С помощью системы компьютерной верстки L^AT_EX и python-библиотеки PyTelegramBotAPI [2] программа заполняет документ, после чего отправляет пользователю правильно оформленный pdf-файл.

Ключевые слова

L^AT_EX, Telegram-бот, Python, верстка, оформление отчета

1 Введение

1.1 Актуальность и значимость проблемы

Десятки научных руководителей ежегодно льют кровавые слёзы читая документацию студентов. Съехавшие страницы, шаблонные ошибки, неправильные отступы и полное непонимание того, что должно быть в документации к курсовому проекту. Хотелось бы облегчить участь учащихся и их менторов.

Студентам было бы очень удобно иметь под рукой платформу, где они могли, не беспокоясь о конечном визуальном облике документа, полностью сосредоточиться на содержании и просто и быстро изложить результат своей работы.

1.2 Описание предметной области

1.2.1 L^AT_EX

L^AT_EX - это система компьютерной верстки, главной идеей которой является фокусировка на содержании самой работы, не тревожась о конечном визуальном облике.

Пользователь L^AT_EX заранее указывает архитектуру документа (добавление глав, ссылок, картинок, таблиц, введение библиографии) с помощью специальных команд. Задача же L^AT_EX – решить вопросы его отображения.

1.2.2 Telegram-бот

Telegram-бот – это специальные небольшие приложения, работающие в мессенджере Telegram предоставляющие интерфейс взаимодействия пользователя с какой-либо программой.

1.3 Постановка задачи

Конечной **целью** проекта являлось создание Telegram-бота, который, взаимодействуя с вводом пользователя, заполнял итоговый отчет курсового проекта, а в конце присылал pdf-файл оформленный по официальным рекомендациям [5].

Интерфейс Telegram-бота должен быть интуитивно понятным, пользователь не должен испытывать каких-то проблем с вводом данных, а также от него не требуется никаких знаний L^AT_EX верстки.

Пользователь вводит всю необходимую информацию, нужную для выбранного шаблона. Например, свое имя, группу, руководителя, библиографию, содержание раздела “Введение”, “Аннотация” и т.д.

При этом бот будет подсказывать, что и как лучше написать, какой раздел отчета как заполнять.

2 Обзор литературы

Идейно задачей проекта является создание такой платформы, где студент мог бы без специальных знаний по форматированию текста, не беспокоясь об официальных правилах оформления, написать работу, сосредоточившись на её содержании.

Основными аналогами этого проекта являются различные текстовые или L^AT_EX редакторы, различные шаблоны, а также онлайн сервисы для улучшения текста.

Редактор MS Word удобен в своём использовании. У него простой, понятный интерфейс. Все элементы управления легко доступны и на виду. Однако файлы, набранные в Word не всегда корректно отображаются (или вовсе не открываются) на другом устройстве. Почти каждый сталкивался со съехавшими страницами, а также проблемным размещением большого количества изображений. Все эти проблемы аналогично возникают даже если студент работает с шаблоном документа.

Можно увидеть и другие шаблоны, набранные в L^AT_EX. Среди его достоинств рубрикация документа, нумерация формул, рисунков и списка литературы. Также тут сложно изменять свойства текста, поэтому пользователь может сосредоточиться на структуре, а не оформлении. Однако для того, чтобы работать с T_EX-файлами, нужны базовые знания L^AT_EX. Фактически, студенту нужно выучить язык верстки, на что нужно время и желание.

Приведенные выше реализации, либо не предоставляют качественную верстку документа, либо же требуют дополнительных знаний обработки документов от студента. Именно поэтому в реализации хочется сделать так, чтобы от студента не требовались знания языка верстки, не беспокоили съехавшие страницы и неправильные отступы. От него требуется лишь изложить свою работу, а все остальное программа берет на себя.

3 Работа с Telegram-ботом

Для работы с Telegram-ботом была выбрана Python-библиотека `pyTelegramBotAPI` [2]. Данная библиотека обладает всеми необходимыми функциями для работы с API Telegram, а также имеет большое количество учебных материалов и объемную документацию.

Для взаимодействия с пользователем, в своей программе я использовал следующие функции:

Отправляет в чат (по id) текст:

```
bot.send_message(chat_id, 'Text')
```

Также позволяет форматировать текст, как html и поддерживает смайлы:

```
bot.send_message(chat_id, '<b>Bold text</b>', parse_mode='html')
```

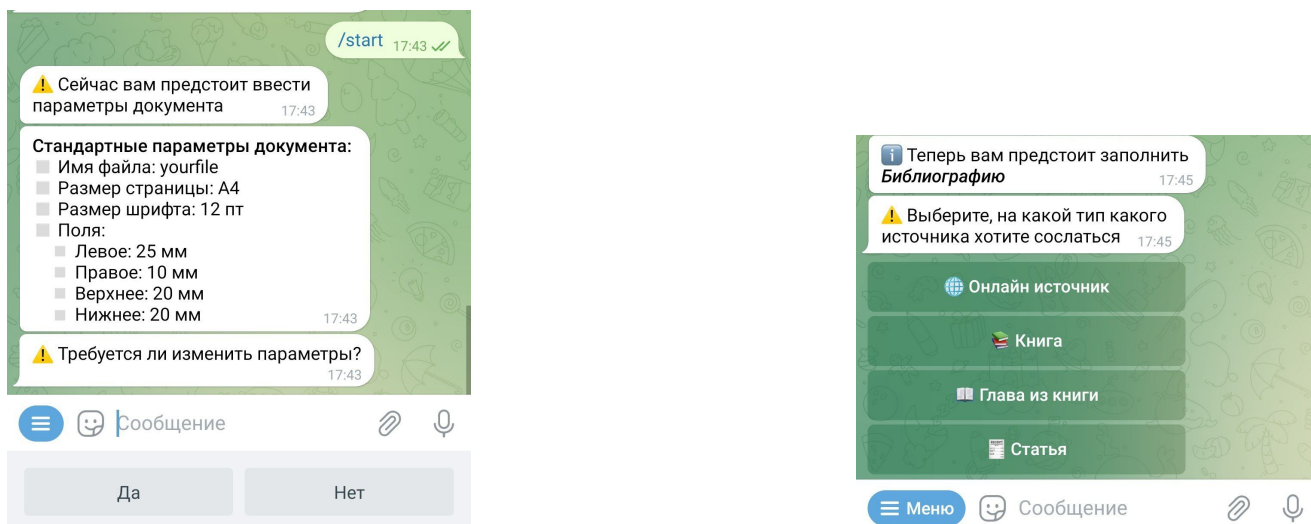
Бот ожидает сообщение (message) пользователя и далее вызывает указанную функцию с аргументом message:

```
bot.register_next_step_handler(message, func)
```

Такой подход с ожиданием конкретного сообщения заставляет писать рекурсивно ввод данных от пользователя. Например, чтобы пользователь ввел таблицу размером 2 на 2, мы должны рекурсивно (а не через for) написать вложенный цикл.

С помощью аргумента `reply_markup` в функции `send_message` можно отправлять пользователю кнопки выбора, как показано на Рисунке 3.1

Рис. 3.1



Также бот поддерживает ввод команд (например /start) при помощи обработчика

```
bot.message_handler(commands=['command'])
```

Ещё присутствует обработчик типа данных сообщения:

```
bot.message_handler(content_type=['type'])
```

Если пользователь отправил сообщение определенного типа (текст, картинка и т.д.), то это сообщение будет обработано в указанной функции.

4 Работа с библиотекой PyLaTeX

Python-библиотека PyLaTeX [1] предоставляет интерфейс взаимодействия L^AT_EX и Python. Она позволяет генерировать T_EX документ посредством python кода.

Основные объекты и функции библиотеки, которые я использовал в своей программе:

- Document() – объект документа; Сюда будет добавляться весь текст.
- Section() – объект секции; Добавляет новую главу в документ.
- Subection() – объект подсекции; Добавляет новую подсекцию в документ.
- Tabular() – объект таблицы; Добавляет новую таблицу в документ.
- Figure() – объект фигуры; Позволяет добавить картинку в документ.
- Command() – объект команды; Добавляет команду в документ.
- Enumerate(), Itemize() – объект нумерованного и маркированного списка; Добавляет список в документ.
- Package() – объект пакета; Подключает пакет.
- NewLine, NewPage – объект команд новой строки и новой страницы; Добавляет новую строку и страницу.
- gen_tex() – Генерирует T_EX-файл.
- append() – Добавляет объект в документ

Используя библиотечные объекты и функции я реализовал подключение необходимых пакетов для успешной генерации T_EX-файла, установку различных параметров документа (отступы, размер, шрифт, размер шрифта и т.д.), добавление текста, картинок, таблиц, простых математических уравнений, разделение документа на главы, генерация титульного листа, добавление библиографии.

5 Работа программы

5.1 Команды

Работа программы проходит таким образом: В начале Telegram-бот отправляет пользователю информацию о себе (что делает, для чего предназначен и т.д.). Далее бот отправляет список доступных команд, нажав на которые, пользователь может начать взаимодействовать с документом.

Доступные команды:

- `start` - Начать;
- `makefile` - Создать документ
- `section` - Добавить главу
- `subsection` - Добавить подглаву
- `subsubsection` - Добавить подподглаву
- `paragraph` - Начать новый параграф
- `table` - Добавить таблицу
- `list` - Добавить список
- `math` - Добавить простое уравнение
- `cite` - Добавить ссылку на библиографию
- `next` - Перейти к заполнению нового раздела

Для добавления текста пользователю не нужно вводить никаких команд, требуется лишь отправить текст. Программа поддерживает жирный шрифт, курсив, подчеркивание и зачеркивание текста, который пользователь набрал в мессенджере.

Аналогично не требуется команда для добавления картинки, пользователь может просто отправить ее.

5.2 Ход работы

Когда пользователь вводит команду `makefile`, бот присылает стандартные параметры документа (имя файла, размер шрифта, поля) и спрашивает у пользователя, требуется ли изменить какой-то из параметров.

У пользователя появляется кнопки выбора “Да” и “Нет”. Если пользователь нажмет на “Да”, перед ним возникнут кнопки выбора параметров, нажав на которые, пользователь сможет начать изменять выбранный параметр.

Когда пользователь закончил вводить параметры документа, ему будет предложено заполнить титульный лист. Пользователю необходимо будет ввести все необходимые данные: ФИО, Тема проекта, Тип проекта, ФИО руководителя, должность руководителя, место работы руководителя.

Заполнив данные титульного листа, пользователя будет предложено заполнить библиографию. Пользователю будет доступен список различных типов источников, на которые можно сослаться. Выбран один из них, он начнет его заполнять. Например, для книжного источника пользователю надо будет ввести имя автора, название книги, год выпуска, издание и серию при наличии.

Аналогичным образом пользователь будет заполнять разделы: Введение, Обзор литературы, Главы, Заключение и т.д. Перед вводом каждого раздела, бот будет подсказывать пользователю что и как лучше написать в данном разделе. Например для *Аннотации* бот выведет подсказку:

По смыслу, аннотация это очень краткий пересказ вашей работы, из которого релевантный человек должен быть способен понять, что вы делали идейно. Она обычно описывает постановку задачи и основные результаты работы в достаточно неформальной формулировке.

5.3 Компиляция PDF-файла

Когда пользователь ввел все необходимые данные для отчета, программа генерирует Т_EX-файл при помощи библиотечной функции `generate_tex()`.

Программа использует компилятор Т_EX-файлов TeX Live [3], Скомпилировать pdf-файл можно, введя эти команды на сервере

```
pdflatex file
biber file
pdflatex file
```

pdflatex file

Скомпилировав pdf-файл, бот отправляет корректно оформленный отчет по курсовому проекту.

6 Заключение

В конечном итоге был реализован Telegram-bot для заполнения правильно оформленного отчета по курсовому проекту.

Удалось реализовать добавление различных объектов в документ, таких как оглавление, таблица, список, математическое выражение, библиография.

В ходе выполнения работы была изучена система верстки L^AT_EX [4], генерация T_EX-документов через PyLaTeX [1], создание Telegram-ботов.

7 Приложения

В репозитории проекта можно посмотреть исходный код, а также увидеть полный пример работы приложения (что присылал бот, что вводил пользователь и какой файл он получил в результате).

Репозиторий проекта – <https://github.com/avabramovv/MJ-Bot>

Telegram-бот – <https://t.me/mjtexBot>

Список литературы

- [1] *PyLaTeX is a Python library for creating and compiling LaTeX files.* URL: <https://jeltef.github.io/PyLaTeX/current/index.html> (дата обр. 10.04.2023).
- [2] *pyTeleBotAPI – synchronous and asynchronous Python implementation for the Telegram Bot API.* URL: <https://github.com/eternnoir/pyTelegramBotAPI> (дата обр. 13.02.2023).
- [3] *TeX Live is intended to be a straightforward way to get up and running with the TeX document production system.* URL: <https://tug.org/texlive/> (дата обр. 13.02.2023).
- [4] Воронцов К.В. *L^AT_EX_{2_ε} в примерах.* URL: <http://www.ccas.ru/voron/download/voron05latex.pdf> (дата обр. 13.02.2023).
- [5] *Методические рекомендации по написанию курсовых проектов для студентов 2 курса бакалавриата программы Прикладная математика и информатика.* URL: <https://docs.google.com/document/d/1Mjhw5jV01bv-XD1PrSyE2nhg8F-z1W9b/edit> (дата обр. 10.04.2023).