

Содержание проекта

Разработать информационно-аналитическое приложение на платформе Windows 10 с Web-интерфейсом для актуального дистрибутива Anaconda (оговаривается с преподавателем).

Функционал и интерфейс

Приложение собирает данные из открытых источников в Интернете по определенной тематике, которая должна быть согласована с преподавателем. Сбор и обновление данных идет по некоторому графику (раз в час, раз в день и т.д.), который может быть настроен и по разовому запросу.

Данные накапливаются в базе данных, управление которой – создание и удаление таблиц, редактирование таблиц доступно пользователю через специальный интерфейс.

На основе собранных данных приложение по графику и по запросу формирует динамические dashboard(ы), которые позволяют анализировать пространственные и временные свойства собранных показателей:

- особенности распределения значений (гистограммы, бокс-плоты, столбчатые и круговые диаграммы),
- динамику изменения показателей
- взаимосвязи (диаграммы рассеивания для количественных показателей, категоризованные бокс-плоты для связи качественных и количественных показателей и кластерные столбчатые диаграммы для связи качественных показателей).

Графические отчеты можно сохранять в виде файлов формата jpeg или pdf.

Dashboard(ы) имеют интерфейс настройки – состав графиков и выбор показателей.

Также доступно построение текстовых отчетов на основе критериев, выбираемых пользователем. Отчеты представляются в виде таблиц с возможностью сохранения в форматах MS Excel и CSV.

Архитектура приложения

Приложение создается на платформе Flask с использованием библиотек DASH и PLOTLY для формирования интерфейсов, графических и текстовых отчетов.

В качестве БД используется MY SQL, для взаимодействия с которой используется библиотека SQLALCHEMY.

Интерфейсы приложения имеют динамические элементы, создаваемые с помощью JavaScript.

Приложение должно запускаться из командной строки запуском на счет основного модуля или командного файла. Установка приложения, в частности создание при необходимости виртуальной среды, создание БД, должна быть полностью автоматизирована, т.е. осуществляться запуском на счет модуля Python или командного файла. Установка приложения и его эксплуатация не должны требовать выполнения системных операций, связанных с изменениями операционной системы, актуального дистрибутива Anaconda и прочих элементов вычислительной среды пользователя. В частности, развертывания ПО Docker и аналогичных инструментов.

Для работы приложения должна быть создана виртуальная среда исключительно средствами менеджера пакетов conda из актуального дистрибутива Anaconda. Таким образом, приложение должно работать в собственной виртуальной среде на любом компьютере, соответствующем требованиям данного документа, без необходимости добавления библиотек к актуальному дистрибутиву Anaconda. Приложение является локальным приложением с графическим интерфейсом, которое работает на изолированном компьютере. Это означает, что для работы приложения не нужны вычислительные ресурсы Интернета и не используется клиент-серверная архитектура (за исключением Web-интерфейса). В тоже время, приложение может анализировать данные из Интернета. Также приложение должно исполняться на компьютере средней мощности с 8Gb оперативной памяти и 2Gb видеопамати не очень мощной видеокарты.

Код разработан в соответствии с требованиями «Python Enhancement Proposals (PEP) 8 -- Style Guide for Python Code» и «PEP 257 -- Docstring Conventions» (см. <https://www.python.org/dev/peps/>).

Документирование

1. Техническое задание с описанием тематики, функционала, интерфейсов и состава библиотек с указанием версий. Также текст ТЗ включает в себя график реализации проекта с указанием дат представления его частей. Высылается заранее для утверждения.
2. Руководство разработчика, включающее описания всех созданных функций и классов и описание структуры каталогов. Представляется вместе с приложением.
3. Руководство пользователя, содержащее инструкции по установке и использованию приложения. Представляется вместе с приложением.

Критерии оценивания

- Работоспособность приложения. При запуске и использовании функционала, указанного в «Руководстве пользователя» не должно возникать ошибок и не должно выводиться в консоль сообщений, связанных с некорректной работой приложения. Возможности языка Python и специально подобранные требования к разрабатываемому продукту позволяют с уверенностью утверждать, что следование указаниям задания для самостоятельной работы гарантирует работоспособность создаваемого при выполнении задания приложения на любом компьютере, на котором установлена общая для всех студентов версия дистрибутива Python — Anaconda, при отсутствии ошибок разработчиков данного приложения. Таким образом, если разработчик корректно создал код приложения оно должно работать, как на его компьютере, так и на компьютере другого пользователя, в частности, преподавателя, при условии выполнения указанных выше требований. Отсутствие возможности приступить к эксплуатации приложения может свидетельствовать только, либо о наличии ошибок в коде, либо о нарушении требований задания, в частности, использовании специфических особенностей компьютера разработчика. Отсутствие возможности приступить к работе с приложением (критическая ошибка¹ при запуске) или прекращение работы приложения по причине возникновения критической ошибки в ходе его эксплуатации автоматически приводит к оценке **ноль баллов по десятибалльной системе**. Наличие не критических ошибок приводит к адекватному снижению балла за приложение.
- Структура приложения. Полнота и качество реализации функционала и интерфейса, описанных в настоящем документе. Приложение должно соответствовать всем требованиям, указанным в настоящем документе. Отсутствие тех или иных требуемых элементов приводит к адекватному снижению балла за приложение. Полное несоответствие требованиям приводит к оценке **ноль баллов по десятибалльной системе**.

¹ Критической ошибкой называется ошибка работы кода приложения, которая делает невозможным его дальнейшую эксплуатацию.

- Интерфейс. Качество реализации и удобство работы с интерфейсом всех частей приложения — минимально возможное наличие ручного ввода данных, интуитивная понятность и удобство размещения элементов графического интерфейса (виджетов), возможность конфигурирования интерфейса (настройки цветов и шрифтов, а также управления окнами с помощью мыши). Минимально допустимой реализацией настройки интерфейса является использование конфигурационных файлов. Приветствуется наличие графического интерфейса для выполнения настроек. Отсутствие возможности настраивать интерфейс является нарушением требований и приводит к оценке **ноль баллов по десятибалльной системе**.
- Реализация кода. Код должен быть разработан в соответствии с требованиями «Python Enhancement Proposals (PEP) 8 -- Style Guide for Python Code» и «PEP 257 -- Docstring Conventions» (см. <https://www.python.org/dev/peps/>). Оценка качества кода в IDE Spyder должна быть не ниже 7 баллов. Более низкая оценка приводит к оценке **ноль баллов по десятибалльной системе**. Более высокий балл является аргументом для повышения оценки. Код должен содержать много комментариев, поясняющих назначение всех его частей. Количество и качество комментариев влияет на оценку.

Документация к приложению. Оценивается качество реализации документации — «Руководства пользователя» и «Руководства разработчика». Документы должны иметь пронумерованные страницы и разбиты на разделы. В документах должно присутствовать оглавление, созданное средствами MS Word. При реализации в формате Adobe Acrobat, оглавление должно обеспечивать навигацию по документу. Технические требования: шрифт Calibri Light 12, межстрочный интервал 1.15, все таблицы и рисунки (скриншоты) пронумерованы и имеют название, каждый документ имеет титульный лист с указанием названия учебного заведения, учебной программы, учебной группы, номера и состава бригады (приведен телефон и адрес Директора).