



## Backend API for the mobile application

Technologies: Python, Django, MongoDB, Ubuntu

URL: <http://www.shopprapp.com>

Duration: 5 months

Business goal of that project was to increase amount of top fashion e-shops like Zara, Mango, H&M, Massimo Dutti that connected to the Shoppr app. We have developed REST API used by mobile application iOS, Android allowing:

- collecting information about new apparels from top fashion websites like Zara, Mango, H&M etc.;
- storing data into MongoDB;
- providing access to the products and performs the user's registration/authorization via REST API.

Among technical issues we solved were parsing the instances with loaded JavaScript and/or JavaScript/AJAX. We also parsed data from the HTML with unspecified rules (e.g. image containing necessary text info, but no mentioning in page/code).



## Automotive Routing Quality Assessment Software

Technologies: Python, MySQL, sqlalchemy, geoalchemy, nosetests, git

Duration: 1 year

Developed a quality evaluation system used to make decisions about possible changes to the routing service. If the quality is found to have dropped, then the changes are not made; if the quality improves, then the deployment of the new data or software affecting the QoS is green-lighted. Aspects of the project included:

- Yandex's MapReduce cluster computations technology
- Statistical modeling, including analytical and computational approaches, bootstrapping, resampling and Monte-Carlo methods
- SQLAlchemy, geoalchemy ORMs



## Industrial facility monitoring application

Technologies: Python, Django, MongoDB

Application monitoring the facility activities in real time and warns users of any delays. Users have capability to close, delay, reschedule and reactivate tasks. Application runs real time function to check if there are any Tasks should be started and monitors each Task which should be started from certain time.

Features:

- several access levels to the data and features
- store data about companies, equipment, operating personnel, all available documents
- runs activation function to start monitoring date and times of the Task
- watches the durations in the steps: step by step according their sequence
- warns users with audio alarm on computer in if Task time is exceeded
- checks if all steps are done in time and warns users that Task has been finished
- marks Task as closed and sets corresponding End Date and Time



## Fantasy Football REST API

Technologies: Python, SQLAlchemy, REST API, py.test

Project Overview: Part 1: Develop a Python wrapper around a 3rd party's REST API that provides NFL game statistics. The list of required API endpoints will be provided. The API provider is Fantasy Data, and they have good documentation and a working test version of their API online (<https://developer.fantasydata.com/documentation>). This part of the project will be open sourced as a general library for working with the Fantasy Data API. Not all endpoints will be supported initially. Only the 3 or 4 endpoints that are needed for the rest of the project will be built.

Part 2: Write three background tasks, preferably Shovel tasks (<https://github.com/seomoz/shovel>), in an existing Python project that synchronizes data between the API and an existing MySQL database. There are already some SQLAlchemy models written, but a couple more models will need to be written as part of this job. These are the three background tasks that need to be written:

- NFL Schedule Synchronizing
- Player Roster Synchronizing
- Post-game NFL Stats Import



## Python Open Source Data Management Library

Technologies: Python

Ambry is a Open Source application for collecting, processing, cleaning and publishing public data. Our goal is to make building a database of hundreds of health, finance and demographic datasets as easy as installing packages with apt-get or rpm.

The software is currently in production, but needs continued development and test. Software is extended with the coverage tests and continued with core development and maintenance.

The documentation, including installation instructions are at <http://ambry.io/>



## Project Room - a Team Management Project

Technologies: Python, Django, jQuery, Google Charts

Duration: 1.5 months

Built a web portal built with the use of Python and Django technologies that was similar to a simplified version of MS Office Project.

Designed the portal so it was distinctly simple to use and accessible from any environment. Utilized Gantt Charts, which allow a visualization of the tasks planned for the project. Implemented interactions within the Charts using the jQuery library so that the planned tasks could be accomplished in order, in a timely fashion, and with the desired effect.



## Logs storage monitoring application

Technologies: Python, Flask, RQ, Redis, JSON, jQuery, WebSockets, REST API, Linux

Duration: 2 months

Monitoring and analyzing of log files with distributed architecture. Software application work in 2 regimes: filters logs for the previous time and monitors them in real time.

Application is able to analyze hundreds of gigabytes of text logs with minimum time delays showing only those logs that meet necessary criteria. Logs could be stored on a distributed basis on different machines in local network.