# Scope

Web application to perform cross tabulation on uploaded csv files.

# Pre-requisites:

Following tools and packages are required to build and test cross tabulation web application:

* Python 2.6
* Django 1.4.18
* Pandas 0.16.0
* MySQL database
* chardet python module

## Note:

Web application was developed and tested on Centos6.6 virtual machine with above tools and packages on Django development server.

# Design Overview:

models.py 

forms.py

views.py

HTML webpage

index.html

MYSQL data base

Database

Frontend webpage

Backend Application

Synchronizes database as per fields specified in models.py. It also provides data to backend for performing cross tabulation on CSV file.

Sends POST request from frontend to backend application. Backend performs required operation on input data and sends result back to the frontend.

## Database:

MYSQL database handles database request from backend application. Stores information for relevant fields specified in models.py file.

## Backend application:

Backend application perform core functionality of uploading csv file, saving the csv file, fetching data from CSV file and database and displaying it on HTML page after performing cross tabulation operation on its data.

### models.py:

Implements “Model” class from Django framework and specifies “FileField” parameter to handle uploading of csv file.

### forms.py:

Implements “Form” class of Django framework and specifies two forms; i) to handle csv upload request and ii) to handle cross tabulation based on options selected in drop down menu field. Forms are important to handle forms on HTML page.

### views.py:

Implements core functionality i.e. all the logic to handle input data posted from HTML page. Main functionality of this module is to;

1. Save the uploaded csv file to directory specified by MEDIA\_ROOT,
2. Fetch data from csv file and display it on HTML webpage
3. Get input from HTML page and perform cross tabulation based on input parameters
4. Display result of cross tabulation on HTML webpage

## Frontend Webpage:

Frontend webpage is a simple HTML webpage that POST’s important input data to backend application and displays data received from backend application to the HTML webpage.

# Steps to execute web application:

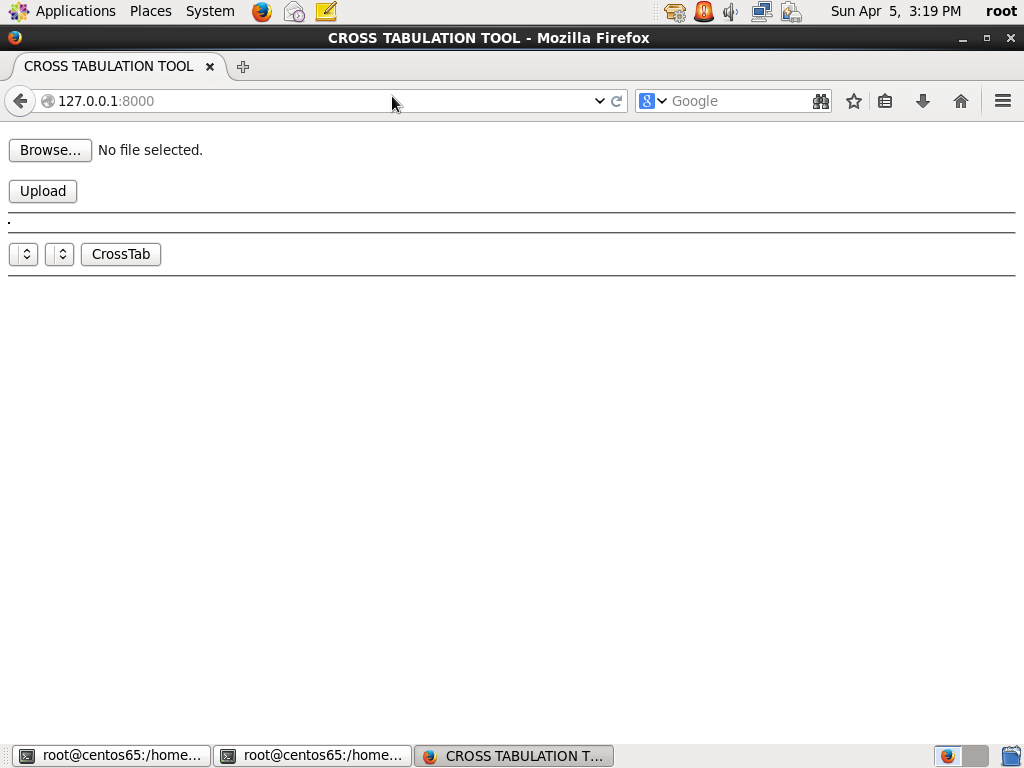
* Start MYSQL database
  + service mysqld start
* Navigate to project directory
  + cd <project\_directory>
* Synchronize webapplication with database
  + python manage.py syncdb
* Start Django development server
  + python manage.py runserver

## ScreenShots:

Please find screenshot for application as below:

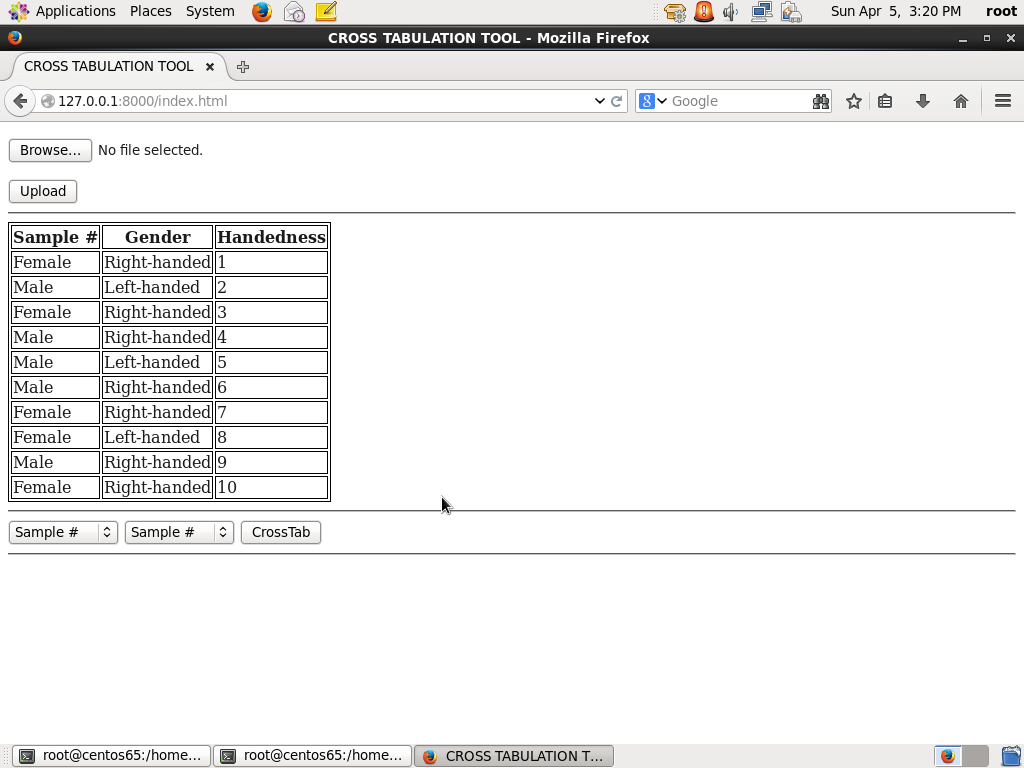
### ScreenShot-1:

Start-up page at http://127.0.0.1:8000



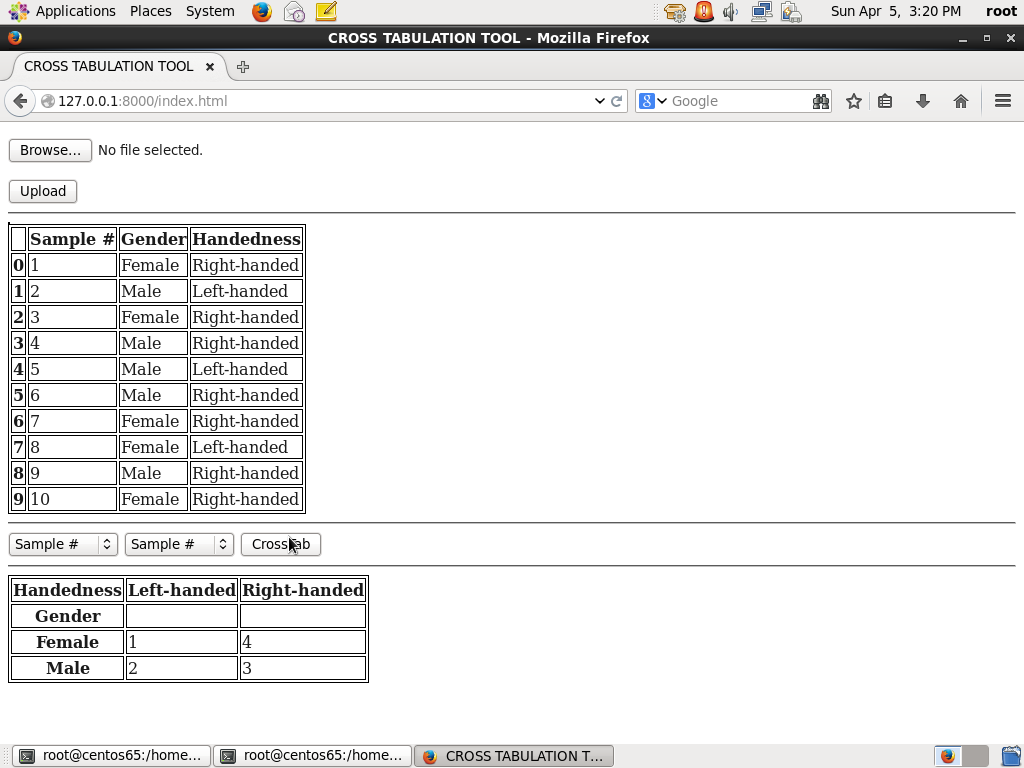
### Screenshot-2

Page after csv file is uploaded by user.



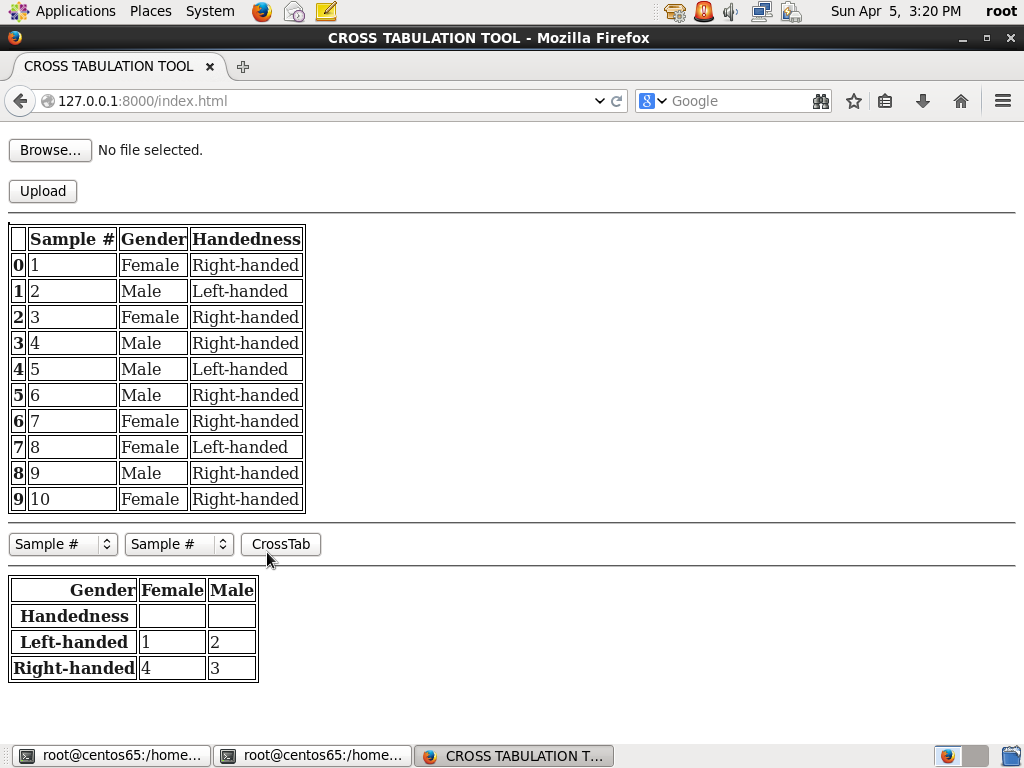
### Screehshot-3:

Page displays cross tabulation performed on CSV file data



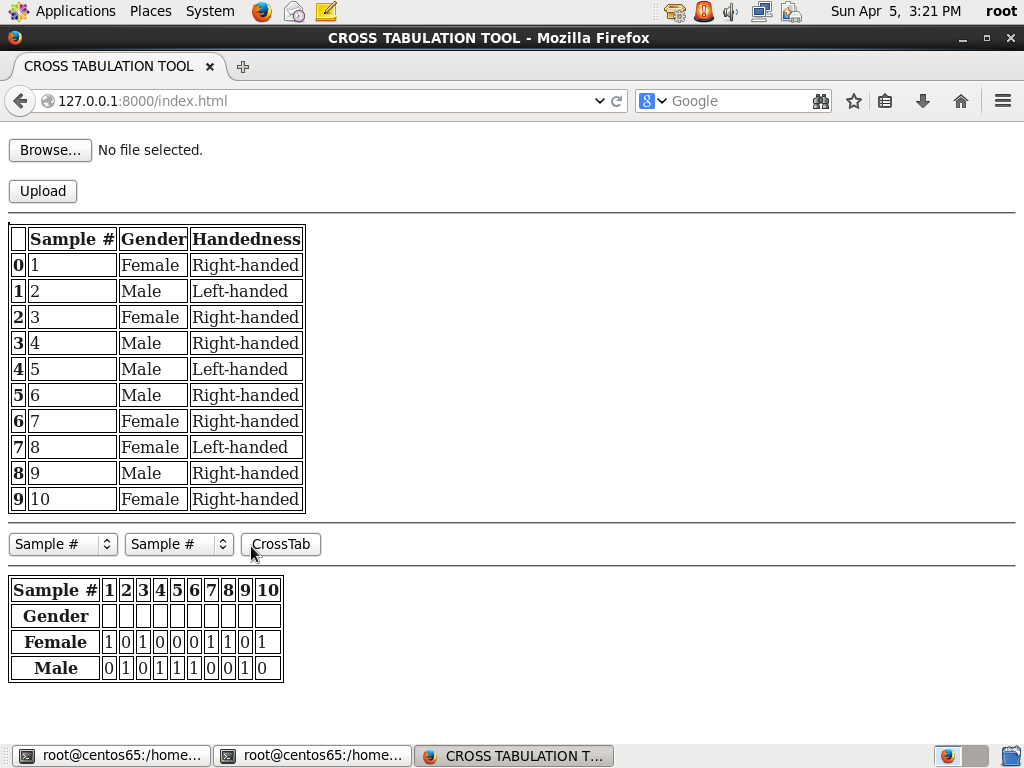
### Screenshot-4:

Some more screenshot for crosstabulation



### Screenshot-5:

Additional screenshot for cross tabulation.



# Design Decisions:

Described below are important design decisions considered while implementing cross tabulation web application:

* CSV storage
  + CSV files are currently stored on disk in directory specified by MEDIA\_ROOT in settings.py
* Fetching data from CSV
  + Data is fetched from CSV files stored on disk. Cross tabulation operation is then performed on fetched data.
* Cross Tabulation
  + Cross tabulation on CSV columns are performed using “pandas”. Its one of the best modules for performing data analysis in python.

# Issues and Constraints:

Priority was given to implement the expected functionality for cross tabulation web application. However, application still has some issues to be handled properly. Following are list of issues and constraints pending to be implemented:

* Display of proper error messages in case user submits form without uploading file.
* Display of proper error messages in case user submits form with same column names for cross tabulation.
* Allowing only CSV file types to be uploaded on web application.

# References:

Creating dynamic website in Django: <https://www.youtube.com/watch?v=bRnm8f6Wavk>

<https://docs.djangoproject.com/en/1.4/>

<https://docs.djangoproject.com/en/1.4/intro/tutorial01/>

[www.stackoverflow.com](http://www.stackoverflow.com)