

AUTOMATING MWR FORM VALIDATION

Bureau of Labor Statistics | Office of Employment and Unemployment Statistics

Eric Kuo, Full Stack Software Fellow | *University of Florida, Computer Science*

Keywords:

python, process automations, web app development

Summary:

In the Data Collection Branch (DCB) at the Office of Employment and Unemployment Statistics, Eric worked on automating the MWR form validation process. Utilizing pdf reading packages with **Python**, Eric was able to extract the text from pdf forms and locate data fields to cross-check against source files. Once the automation scripts were finished, everything was packaged into an easy-to-use web app using **Flask** and **HTML/Jinja/CSS**. This automation web app can save the DCB team **24 total hours** of manual review time **8 times** a year.

Automating MWR Form Validation

Offices of Employment and Unemployment Statistics
QCEW, Data Collection Branch
Bureau of Labor Statistics
Kelly Quinn — Supervisory Economist

coding it forward >



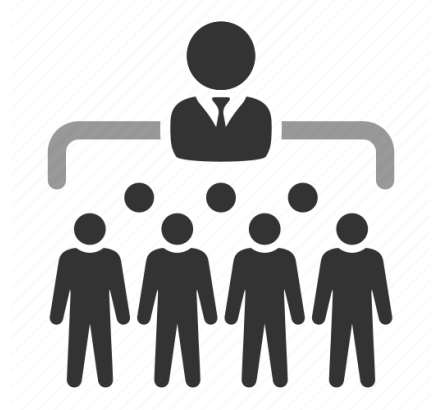
Eric Kuo
University of Florida
Computer Science

AGENDA

1. What are MWR Forms
2. Current Process For Validation
3. Automating the Process
4. Web App Demo
5. Next Steps / Maintenance

MULTIPLE WORKSITE REPORT

- Collects employment and wage data from multi-location employers
- Shows the distribution of employment in different industries within each state
- Useful for analysis of business and economic conditions in geographic areas
- Data ensures equitable allocation of Federal funds
- MWR forms printed and mailed by a third-party contractor



CURRENT VALIDATION PROCESS

- Manually validated **8 times** per year
- DCB coordinates team of reviewers and assigns states for them to review within 24-hour window
- Not all pages can be checked manually (~2,000 pages)
- Each reviewer randomly samples forms from their state
- Potential human error

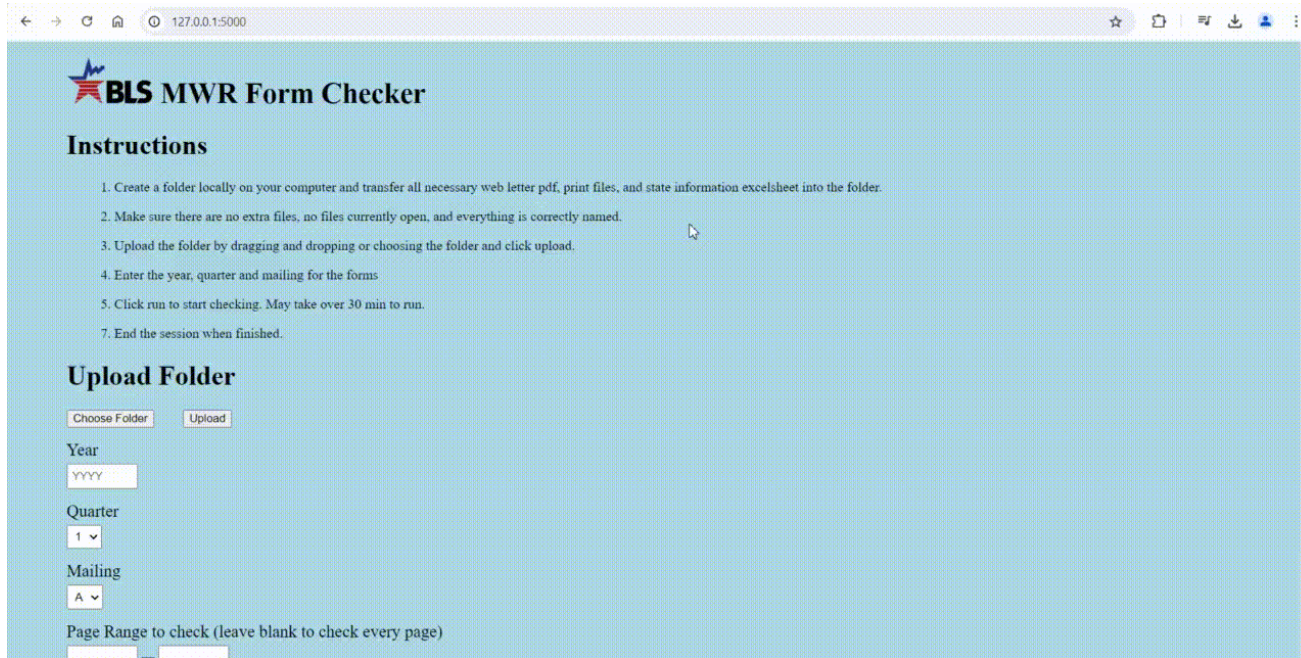
AUTOMATING THE PROCESS

- Used Python to extract text from MWR pdf forms and check if it matches with the source files.
- To locate different data fields, search for keywords. Ex:
 UI number: 1234567890
- Can check every page accurately in ~30 minutes
- Packaged everything into a web app using Flask for the team to use

When there is a task that can be done in 24 hours, but you find a way to automate it in 9 weeks.



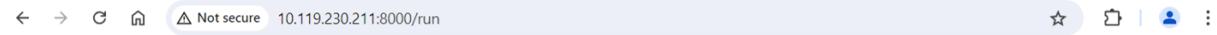
WEB APP




The screenshot shows a web browser window with the address bar displaying "127.0.0.1:5000". The page title is "BLS MWR Form Checker". The main content area has a light blue background and contains the following sections:

- Instructions**: A list of seven steps for using the application, including creating a folder, uploading files, and checking the forms.
- Upload Folder**: A section with two buttons, "Choose Folder" and "Upload", and input fields for "Year" (with a placeholder "YYYY"), "Quarter" (with a dropdown menu showing "1"), and "Mailing" (with a dropdown menu showing "A").
- Page Range to check (leave blank to check every page)**: A text input field.

WEB APP



Results (5)

 Download Report

End Session

State Information Errors (2)

Section 1 Errors (0)

Section 2 Errors (3)

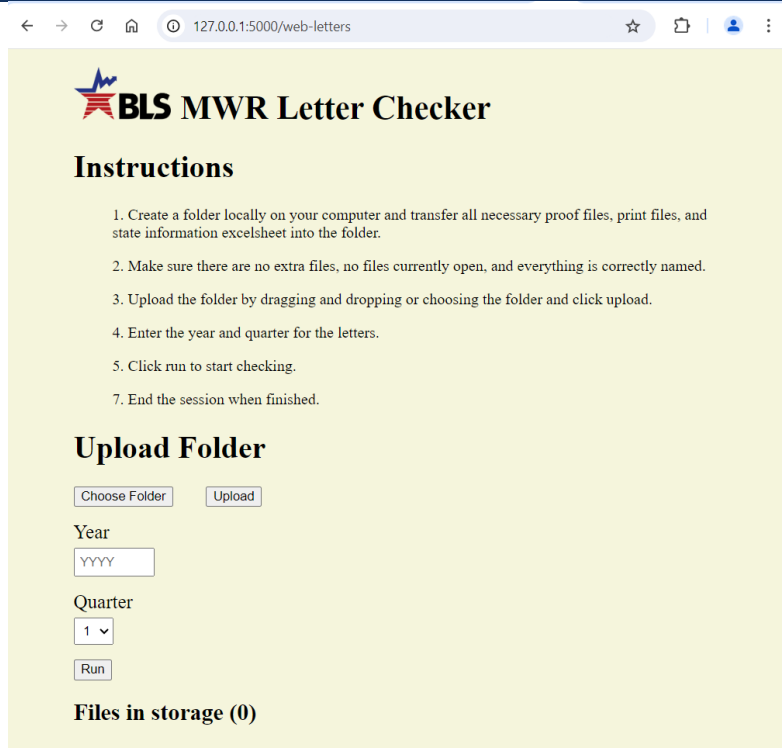
Section 3 Errors (0)

Instructions Errors (0)

Letter Errors (0)

ADDITIONAL WEB LETTER VALIDATION

- Web letters sent to employers to inform them of reporting MWR forms online
- Also checks for some of the same fields such as due date, UI num, addresses, etc.
- Used same process to check information as MWR forms



The screenshot shows a web browser window with the address bar displaying "127.0.0.1:5000/web-letters". The page has a light yellow background and features the BLS logo (a red star with a blue wave) and the title "BLS MWR Letter Checker". Below the title is a section titled "Instructions" with a list of seven steps: 1. Create a folder locally on your computer and transfer all necessary proof files, print files, and state information excelsheet into the folder. 2. Make sure there are no extra files, no files currently open, and everything is correctly named. 3. Upload the folder by dragging and dropping or choosing the folder and click upload. 4. Enter the year and quarter for the letters. 5. Click run to start checking. 7. End the session when finished. Below the instructions is a section titled "Upload Folder" containing two buttons: "Choose Folder" and "Upload". Under these buttons are input fields for "Year" (containing "YYYY") and "Quarter" (a dropdown menu showing "1"). Below the "Quarter" dropdown is a "Run" button. At the bottom of the form, it says "Files in storage (0)".

Web Letter Page

NEXT STEPS / MAINTENANCE

- Web App is deployed and ready to use for the team
- Potential issues if format of pdf forms ever changes
- Deployed web apps require someone to maintain it. For example, if there is an exploit discovered in one of the Python packages.
- New features requested by team
- Optimizing for faster runtime

THANKS FOR COMING

Acknowledgments:
Coding It Forward,
Supervisors Kelly, Joyce and Sharon,
Erin and Brandon