Saving Data Journalism II: Building a production-level system to preserve dynamic websites

Statement of National Need

Data journalism stories are among the most innovative and original works being produced by newsrooms today. Iconic examples include "Mapping Segregation" by *The New York Times* and "Are Hospitals Near Me Ready for Coronavirus?" by ProPublica. Projects like these, created by news organizations in dozens of countries, are custom-built websites that display content dynamically in the browser. The focus of these stories, in keeping with the larger missions of journalism, is to serve underserved communities, expose inequalities in society, and give voice to the concerns of people from all backgrounds. During the COVID-19 pandemic they have taken on a compelling gravity, as they help us understand the spread, severity, and impact of the coronavirus. Yet because of their technological complexity, these websites, often referred to as "news apps" or interactives, cannot be fully or systematically captured by current web archiving tools. Newsrooms are also often under budgetary and resource constraints that limit archiving efforts. As a result, they are disappearing.

Libraries have the mandate, the expertise, and the infrastructure to save these works for the long term. In a 2018-2019 planning grant funded by the IMLS (LG-87-18-0062-18), NYU Libraries built the first-ever emulation-based web archiving prototype capable of capturing the look, feel, and functionality of a database-reliant news app. ReproZip-Web is an open-source web archiving tool that creates a bundle containing all the information needed to reproduce a news application, and its lightweight nature makes it ideal for distribution and preservation. However, further work is needed to make it operational in newsrooms, where the archiving process must begin.

To this end, NYU Libraries, in collaboration with the NYU Visualization and Data Analytics Center; Webrecorder, an independent organization developing open-source web archiving tools; ProPublica; and the *Los Angeles Times*, requests $249,905 for a project to bring this prototype into production to enable the capture of these dynamic websites, at scale, for long-term archiving and preservation. This project would be categorized in the IMLS scaling phase of maturity.

Project Design

Web archiving technology was built around the static web, and its tools are not able to capture the look, feel, and functionality of dynamic websites. To save these applications for the long term, we must go beyond static crawls and capture the application with its computational environment and data, so as to make it accessible via emulation. Our open-source prototype captures dynamic websites by leveraging ReproZip, a computational reproducibility tool, and Webrecorder, an open web archiving tool. The tool produces a bundle (.rpz) suitable for long-term preservation, which years from now could be opened and unpacked to emulate a website produced today. It contains all of the dependencies needed, including the precise information and source code for the computational environment (e.g., the operating system, software libraries) as well as all data and files needed to replay the captured site. By successfully preserving ProPublica's database-reliant news application "Dollars for Docs," we demonstrated that our prototype can archive and reproduce an entire news application.

This project has two larger streams of work: 1) technical improvements and an expansion of ReproZip-Web and 2) user testing with newsrooms to improve workflow compatibility and encourage adoption of the tool as well as build awareness of the need to archive data journalism projects. Initially, our team will use a survey to establish a baseline of newsrooms' data journalism production workflows. This will inform fundamental development decisions of ReproZip-Web and help us conceptualize where the tool might naturally fit into their publishing process. We will then solicit one data journalism app from each newsroom, and use these examples to build out the prototype for testing on a range of applications. After approximately nine months of development, we will ask

newsrooms to test out the tool and capture one of their projects with ReproZip-Web in the first of the recorded UX/working sessions, during which we will do a deeper dive into each newsroom's workflow. After each session, we will use the detailed data, input, and feedback we have gathered from users to inform the technical work, which will be highly iterative as we run into further edge cases and requirements.

The technical development will be led by two software engineers uniquely qualified to solve this problem — Rémi Rampin, a lead developer on ReproZip, and Ilya Kreymer, the developer of pywb and Webrecorder. Their work will focus on putting ReproZip-Web into production by maturing the tool and improving the scale and fidelity of the web archiving. The project plan in months is as follows:

1-3	Project Manager (PM)/PIs: Hire a graduate assistant. Survey newsrooms on their production workflow and secure participation from five organizations. PM and Graduate Assistant (GA): Request news apps for testing.	**13-15**	GA: Analyze and report out feedback. Dev: Incorporate feedback, iterate and develop; look into issues of data stored remotely (e.g., assets stored on Amazon S3 buckets). PM: Schedule second round of online UX/working sessions.
4-6	GA: Secure news apps for testing. Development team (Dev): Build out ReproZip-Web (RZW) to support other technologies and languages, handle more edge cases and errors, and optimize for code space and speed of execution.	**16-18**	Full team: Conduct second round of UX/working sessions. Dev: Continue to incorporate feedback, iterate and develop; updates to optimize ReproServer for replaying RZW packages in the cloud.
7-9	Dev: Further integration with the Webrecorder tool set: on-demand archiving for capture of interactive client-side content, Webrecorder Autopilot for automating types of capture. PM: Schedule first round of online UX/working sessions.	**19-21**	GA: Analyze and report out feedback. Create technical and user documentation. Dev: Continue integration with Webrecorder for use of remote browser system for capture and replay, upgrade pywb. Finalize the user interface.
10-12	Full team: Conduct first round of UX/working sessions.	**22-24**	PM/PIs: Branding, dissemination, and promotion.

National Impact

By further developing our prototype, we will be able to significantly increase the coverage and adoption of the tool to save data journalism projects at scale. But the impact of this project has the potential to extend well beyond news applications. Database-reliant journalism stories are but one example of sophisticated digital projects; other potential use cases include digital humanities projects, digital museum exhibits, and newer forms of interactives. Our prototype, an open-source tool that can easily pack and replay dynamic digital objects, is the first of its kind, and has the potential to save thousands of digitally-born works. Its interoperability with Webrecorder and the established history of development and support for ReproZip make it a sustainable solution to preserving these fragile works.

Budget Summary

The requested funds of $249,905 would cover $55,834 in personnel (40% of one research engineer salary for an 18-month term) with 30% fringe benefits of $16,750; $15,750 for a graduate research assistant with 30% fringe benefits of $4,724; $74,391 for a software engineer contract; $8,000 for web design and branding; $1,900 for conference travel; and $12,260 in other costs for participant fees for working sessions, UX software tools, and transcription costs. Direct costs will total $189,609. Indirect costs will total $60,296 at a negotiated rate of 31.8%. Cost share is not applicable for this project.