Original software publication

New, flexible and open-source fisheries self-reporting app: The Shiny4SelfReport

Eurico Mesquita Noleto-Filho [a,*], Ronaldo Angelini [b], Jeroen Steenbeek [c], Adriana Rosa Carvalho [a,d]

[a] Triatlas Project, Department of Ecology - Federal University of Rio Grande do Norte, Campus Universitário Lagoa Nova. Code 59078-900, Natal RN, Brazil
[b] Department of Civil and Environmental Engineering - Federal University of Rio Grande do Norte, Campus Universitário Lagoa Nova. Code 59078-900, Natal RN, Brazil
[c] Ecopath International Initiative, Barcelona, Spain
[d] Fishing Ecology Management and Economics (FEME), Universidade Federal do Rio Grande do Norte – UFRN. Depto. de Ecology, Natal, RN, Brazil

ARTICLE INFO

ABSTRACT

Self-reporting applications are considered a promising solution for fisheries data monitoring. However, they are still failing in providing accurate information and engaging users. We introduce the Shiny4SelfReport, an application for self-reporting data in fisheries that aims to address these shortcomings. Instead of using expensive proprietary software, we demonstrate how common and affordable technologies can be used to fill gaps in fisheries management of developing nations. The tool, developed in R, works by gathering fishers' inputs and storing them in the cloud. It was designed to be simple and adaptable. Our application improves the data assembly on small-scale fisheries, provides fishers' engagement and data accuracy, and may integrate fisheries' knowledge worldwide while filling the gaps on data-poor fisheries. The app is available at http://triatlas.shinyapps.io/Shiny4SelfReport under a General Public License (GPLv3).

Code metadata

Current code version	1.0
Permanent link to code/repository used of this code version	https://github.com/ElsevierSoftwareX/SOFTX-D-21-00079
Legal Code License	GNU General Public License (GPLv3)
Code versioning system used	git
Software code languages, tools, and services used	R, js
Compilation requirements, operating environments & dependencies	R
If available Link to developer documentation/manual	None
Support email for questions	euriconoleto@hotmail.com

Software metadata

Current code version	1.0
Permanent link to executables of this version	https://triatlas.shinyapps.io/Shiny4SelfReport
Legal Software License	GNU General Public License (GPLv3)
Computing platform/Operating System	Linux, OS X, Microsoft Windows, Unix-like, distributed/web based.
Software code languages, tools, and services used	R, js
Compilation requirements, operating environments & dependencies	R
If available Link to developer documentation/manual	https://github.com/Filhomn/Paper/blob/main/Shiny4SelfReport-main/User%20Guide%20.pdf
Support email for questions	euriconoleto@hotmail.com

---

* Corresponding author.
  E-mail address: euriconoleto@hotmail.com (Eurico Mesquita Noleto-Filho).

Eurico Mesquita Noleto-Filho, Ronaldo Angelini, Jeroen Steenbeek et al.

SoftwareX 16 (2021) 100843

1. Motivation and significance

The approximately 10,000 fisheries worldwide [1] are thought only to meet sustainability goals if adequately managed [2,3]. However, fisheries management is generally limited by poor quality or insufficient data, lack of monitoring, and the lack of resources to carry out data collection to support management [4]. Currently, only 6% of world fisheries are systematically assessed and managed, and approximately 25% are unreported [1].

In fisheries research, there is an increasing need to diversify data sources as an alternative for formal surveys, which are expensive and difficult to execute, especially in developing countries. Lacking proper surveys, researchers tend to use logbooks and reports as alternative data sources [5].

The increased popularity of smartphones has provided the scientific community with the means to substitute paper logbooks with self-reporting applications. Many smartphones have a digital barometer, an altimeter, a magnetometer, an ambient light sensor, an accelerometer, and a gyroscope, which are useful for scientific purposes [6], especially in environmental and fisheries issues.

There are several applications for self-reporting catches, fishing effort, and prices already available, such as iAngler [5], eCatch [7], OurFish (https://rare.org/story/tracing-fish-and-finances), Deckhand (http://deckhandapp.com), mFish (https://mfish.co), FACTS (https://www.fisheryfacts.com), FishBrain (https://fishbrain.com), Tails n'Scales (https://tailsnscales.org), Fishface (https://www.natureaustralia.org.au/what-we-do/our-priorities/oceans/ocean-stories/fishface), MyFishCount (https://www.myfishcount.com), and Abalobi (https://abalobi.info). However, these apps do not cover every type of fishery and fishers because they are only available in a limited number of languages and include only a limited number of species. This prevents many fisheries in predominantly lower- and middle-income nations from engaging with such apps.

Open-source languages and software are emerging as a central resource for most technological advancements as open-source software for technical computing in government, academia, and industry [8]. Unlike closed-source software, they are flexible and customizable, allowing the developers to modify the code for free [9]. None of the existing self-reporting apps are open-source, which hampers researchers from adapting them to meet local needs. However, there are still many researchers without any programming knowledge, to whom adapting open-source applications may be a challenge. In the last few years, the need for programming knowledge for using open-source languages and software, the growing demand for customized applications, and the shortage of qualified developers has boosted the adoption of "no-code" platforms. For researchers without any programming knowledge, a no-code platform may be the only alternative to customize software.

In the scientific community, two of the most used programming languages are R [10] and Python [11], with R being the most popular among scientists given its vast statistical libraries. As such, we have developed Shiny4SelfReport (SSR), an open-source application written in R for creating self-reporting applications in fisheries and environment research. Shiny4SelfReport is designed for adaptability, where researchers only have to change a few lines of code to shape the app to local assessment needs. For users that do not know how to code, Shiny4SefReport provides a no-code interface, allowing full modifiability without the need for programming.

2. Software description

The Shiny4SelfReport (SSR) framework consists of two interfaces: the User Interface used by end-users for data entry and the Developer Interface, where SSR is tailored to specific ecosystems and management needs. These interfaces connect to a remote server, as described in detail below.

The SSR framework requires the rdrop2 and shiny R packages. SSR and its source code are provided free of charge at <http://triatlas.shinyapps.io/Shiny4SelfReport> under a General Public License (GPLv3).

2.1. User interface

The user interface consists of two tabs: Questions and About (Fig. 1). In the About section, users learn more about the authors, the application, and server Terms of Use. In the Questions section, the fishers enter catch information that will be stored in the cloud.

The first version of SSR, which was inspired by the need to collect data from small-scale fisheries in order to support management in Brazil, requires information about the fisher's identification, the catch by species in kg, and the effort in days. We designed the application to be used on small and moving vessels, where accurately typing on the phone is difficult and time to enter data is short due to smaller crew sizes that handle various tasks. As such, we used sliders for entering numeric values (Fig. 2) to facilitate usability, and photos can be used instead of species names for illiterate fishers.

2.2. Developer interface

The developer interface is where one customizes the inputs of SSR (Fig. 3). Native no-code support ensures that no R programming knowledge is needed. By default, the Developer interface is shown when one logs into the app with the User ID "developer" then clicking 'About' (Fig. 3A). This access user ID can be changed if needed.

The Developer interface is divided into two tabs: Inputs and Code. The Inputs tab allows customizing which captions and inputs must be available in the User interface (Fig. 3B). Developers can change letter type, the position, and the number of inputs available (sliders, text, and file). Texts can be entered in any language and can use local species names. Questions and the units of inputs can be adjusted to local needs. Last, data management and storage can be directed to any cloud account, thus enhancing privacy and security by being independent of the app's developer. The Code tab exports a code written in R, which contains all the customizations made in the Developer interface. The user has to save the code in a folder, after which the modifications are integrated into the Shiny4SelfReport framework.

2.3. Connection with the cloud

Cloud storage is used to collect user inputs from multiple connected devices. The default cloud package used is rdrop2, which allows connecting the app directly to a Dropbox account (www.dropbox.com). After configuring the app (Fig. 4A), a login page from Dropbox will appear automatically (Fig. 4B). The user must create his/her account to connect to the app. All the data can be analyzed directly in R software by using the rdrop2 package, or the users can download the data and analyze it in the software of their preference.
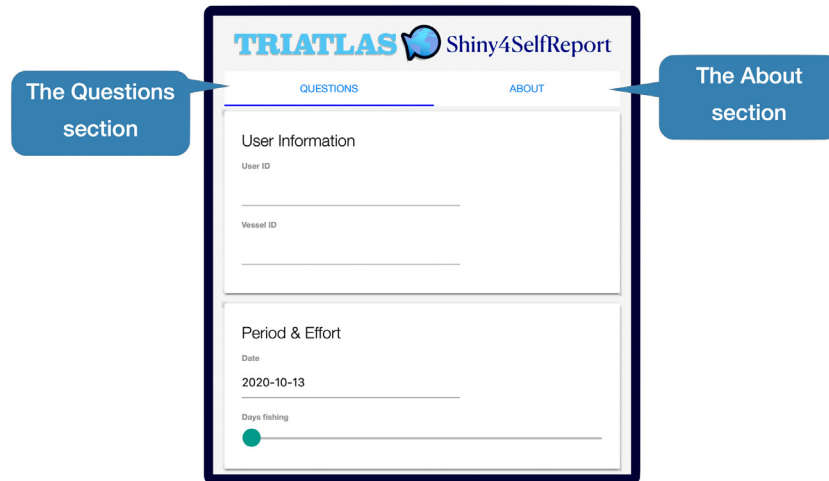
Fig. 1. The Shiny4SelfReport user interface is divided in two tabs: Questions and About. At the **Questions tab,** users can input information about their fisheries (e.g. Photos, Catch in kg, Prices), and upload to the server. At the **About tab**, users learn more about the authors, the TRIATLAS project, and the application and server **Terms of Use**.
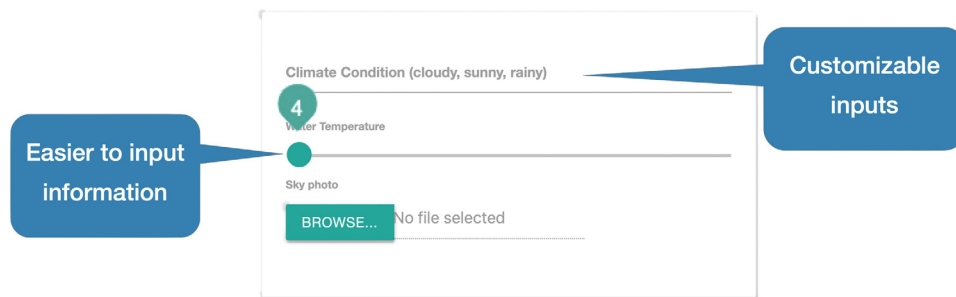

Fig. 2. The app has simple and straightforward inputs which does not require the fishers to type. Also, Shiny4SelfReport is flexible allowing the researchers to customize its inputs to new kinds of surveys and samplings strategies.

2.4. Compatibility with progressive web application technology

Shiny4SelfReport is available as a Progressive Web Application (PWA) (Fig. 5), which installs as a native app on mobile devices. The method to install a PWA may vary depending on the operation system, but it is usually similar to the described above (Fig. 5). The PWA technology works in both IOS and Android systems.

3. Implementation and impact

Shiny4SelfReport provides researchers with an easy framework for creating self-reporting applications. Even though designed for fisheries, it can be used to monitor and gather data for any other application situation. Researchers can adapt the Shiny4SefReport app to different aquatic or terrestrial extractive economic activities or other fields (including laboratory procedures), surveys, and sampling strategies. SSR facilitates user engagement by its design for usability and flexibility, using any spoken language, using photos instead of species names (for illiterate fishers), and displaying only the species and variables of interest.

Last, any researcher can customize the tool using any device – smartphone, tablet, or computer – with Internet connectivity.

3.1. Applicability

Due to its simple interface built for easy use, SSR could be helpful in a range of fields and disciplines. For instance, in oceanography, Shiny4SefReport could benefit animal surveys, report abiotic variables (phosphorus, nitrogen), and report climate conditions (temperature, wind, waves). In biology, it may assist in data gathering on cellular biologies, such as the number of blood cells or even reporting patient information. The Shiny4SefReport app could also help gather data in election polls. To allow any of the uses mentioned above and others, researchers can modify the app by changing the code directly in R or using our no-code interface within the app.

As the tool is a ready-to-use technology with the potential to obtain (temporal or spatial) large-scale standardized data, it enables the implementation of self-monitoring by a wide range of players. In many world-fishing areas, markedly in developing countries, production is poorly registered due to the lack of human and financial resources to track landings at distant and diverse ports [12,13]. Even if resources are available, fisheries' varied and complex operation and structure [14,15] result in scattered monitoring and incomplete data to inform management. Production is still unaccounted for mainly in small-scale fisheries because catches are not recorded by any tax payment [16]. Most of the catches can be consumed locally by dispersed trading [17].
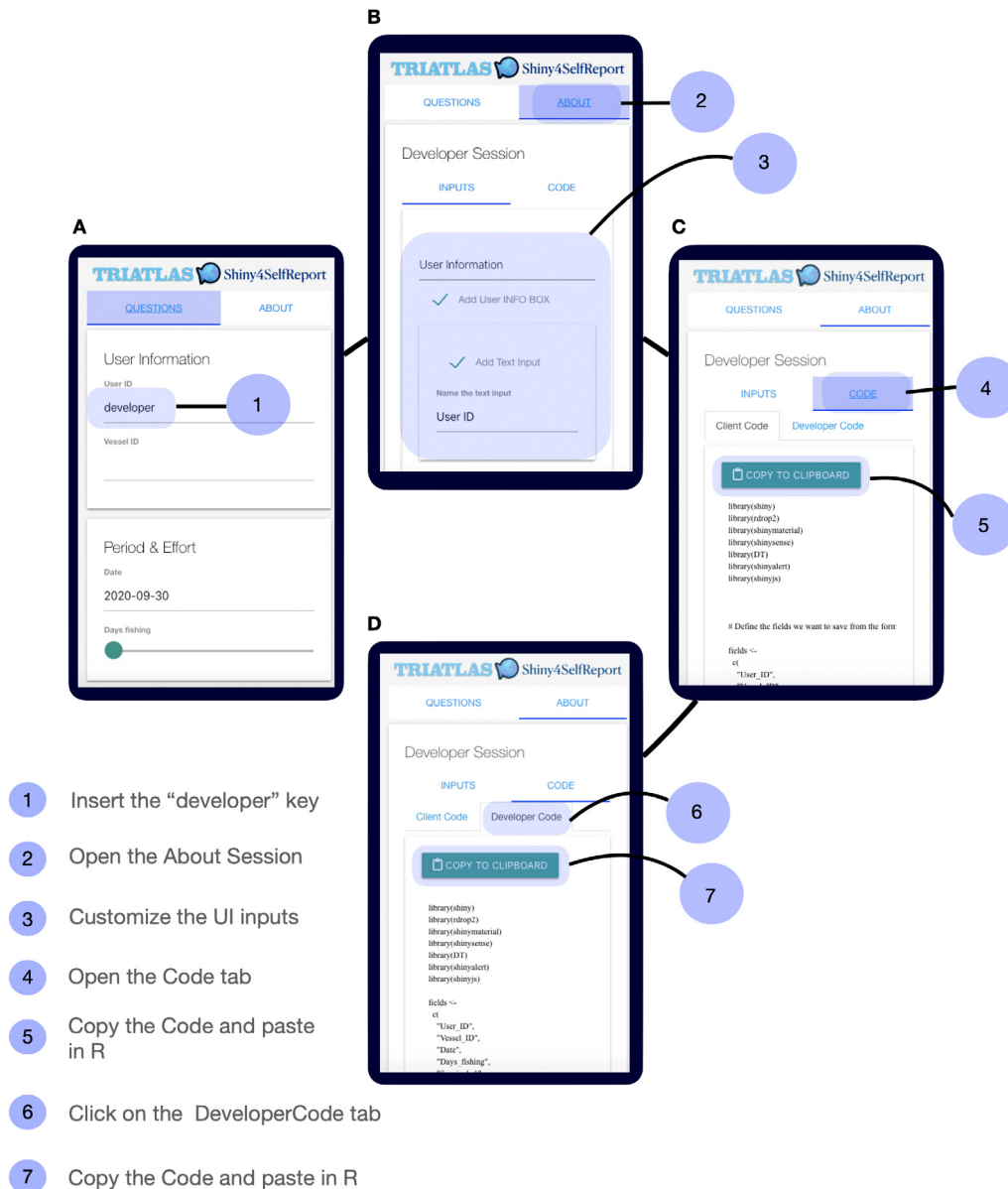
Eurico Mesquita Noleto-Filho, Ronaldo Angelini, Jeroen Steenbeek et al.

SoftwareX 16 (2021) 100843

Fig. 3. Accessing and using the developer interface step by step. **A:** inserting the key in the User ID; **B:** Opening the About section and customizing the inputs in the INPUT tab; **C:** Exporting the code to use in R; and **D:** Exporting the code to download the data from the cloud.

Through the app presented here and by using cell phones, any fisheries player could regularly provide data gathering. Thus, as envisioned here, fishers may easily self-monitor their catches daily and help unveil trends not informed by the conventional data sampling steered by researchers and managers. In addition, the app may enlighten information on production, dependence on spawning aggregation, fishing effort, or any other information elected by the stakeholders, improving the understanding of each sector (industrial and small-scale fishery).

Beyond involving fishers in data gathering, the app spotlights their participative potential. It gradually makes them aware of the data reported, providing them compelling information for entering into fishing-related decisions. Likewise, competent authorities and managers will identify the potential of actively informed fishers to engage in data collection and managerial

insights discussion. Currently, the app is being applied in fishing communities in three districts in Brazil, and its code has been adapted for use in five fishing communities in Cabo Verde (Africa).

These first uses will work as a validation procedure to confirm our premise of its flexible and straightforward routine. Also, it will be the opportunity for fishers to self-monitor their catches through the app, becoming part of data generation for establishing public policies and management decisions.

3.2. Privacy

Instead of centralizing all user's data within a proprietary cloud storage, the SSR tool uses third-party cloud services, like Dropbox, that are assigned when the app is customized. Hence, we address Security and Privacy concerns by allowing the data
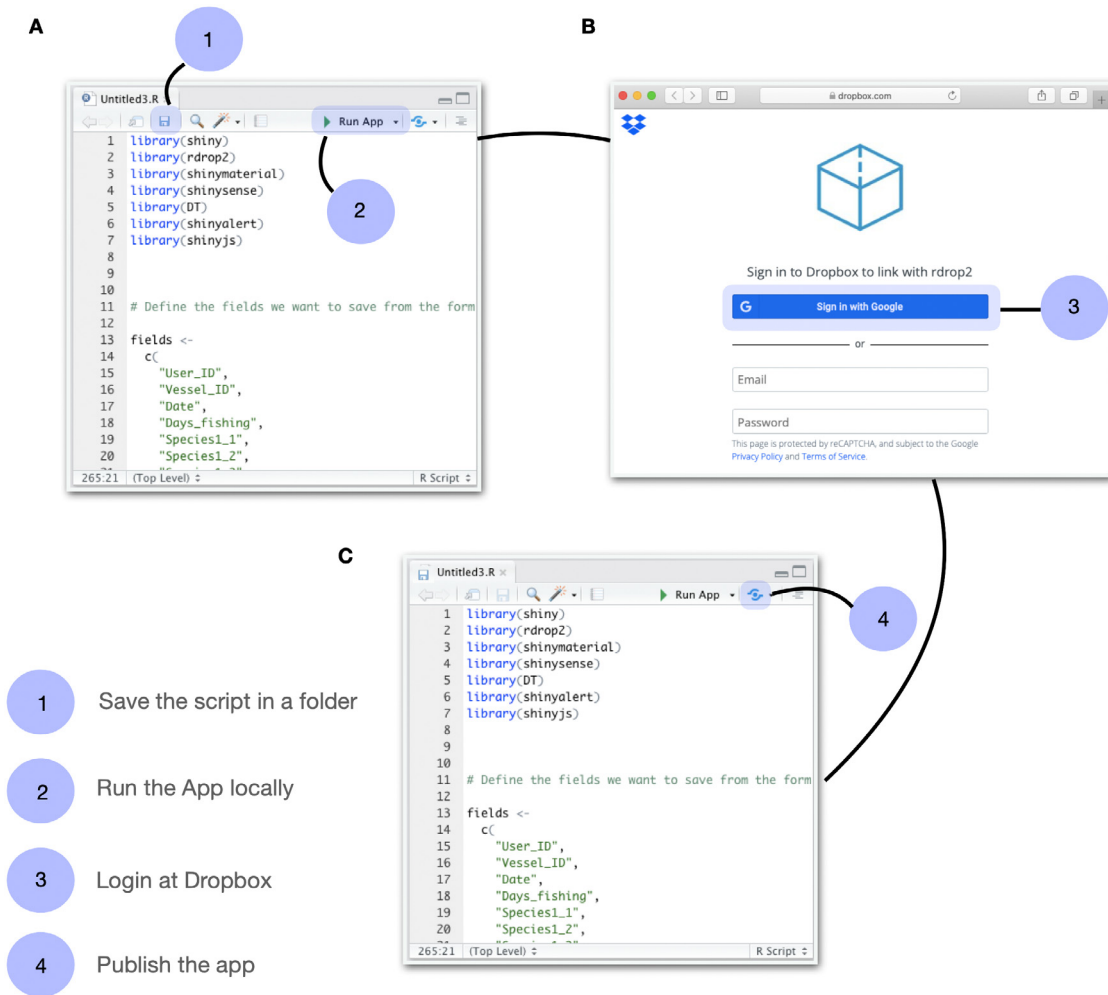
Eurico Mesquita Noleto-Filho, Ronaldo Angelini, Jeroen Steenbeek et al.

SoftwareX 16 (2021) 100843

Fig. 4. Deploying the code on the web step by step. **A:** Save the script in a folder and run it locally; **B:** A window will appear so you can connect the app to your Dropbox account; **C:** Publish the app on the web so fishers can use it on their smartphones.

to be stored in a well-known cloud service with a mature privacy policy. The information is stored directly in the user's account. By no means can the SSR development team access collected data.

4. Conclusion

In this study, we present a flexible and open-source tool for building self-reporting applications. Instead of using expensive proprietary software, we demonstrate how common and affordable technologies can fill gaps in fisheries management of developing nations. Also, the SSR approach shows promising utility beyond fisheries management. It is helpful to register a wide sort of data surveys in many scientific areas like chemistry and physics.

Credit authorship contribution statement

Eurico Mesquita Noleto-Filho: Conceived the methodology, Wrote web R application, Wrote the application documentation, Tested the software, Contributed critically to the draft. **Ronaldo Angelini:** Conceived the methodology, Wrote the application documentation, Tested the software, Contributed critically to the draft. **Jeroen Steenbeek:** Conceived the methodology, Wrote the application documentation, Tested the software, Contributed critically to the draft. **Adriana Rosa Carvalho:** Conceived the methodology, Wrote the application documentation, Tested the software, Contributed critically to the draft.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.
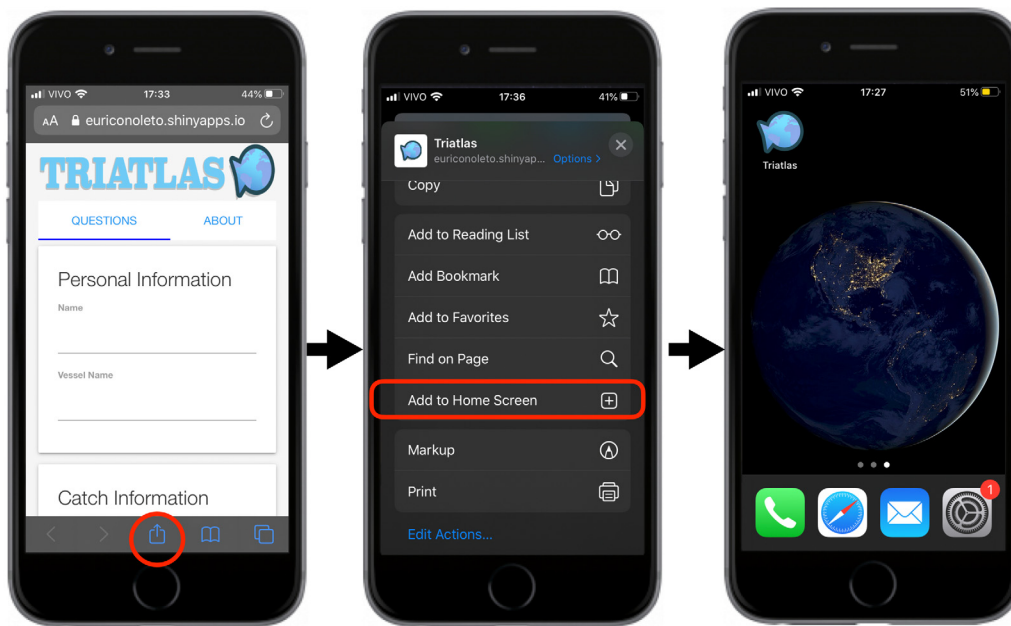
Fig. 5. Shiny4SelfReport install using Progressive Web Application technology.

References

[1] Fujita R, Cusack C, Karasik R, Takade-Heumacher H, Baker C. Technologies for improving fisheries monitoring. San Francisco; 2018, p. 71.

[2] Costello C, Ovando D, Clavelle T, Strauss CK, Hilborn R, et al. Global fishery prospects under contrasting management regimes. Proc Natl Acad Sci 2016;113(18):5125–9, Available from: http://www.pnas.org/lookup/doi/10.1073/pnas.1520420113. [Internet].

[3] Gaines SD, Costello C, Owashi B, Mangin T, Bone J, Molinos JG, et al. Improved fisheries management could offset many negative effects of climate change. Sci Adv 2018;4(8). eaao1378, Available from: https://advances.sciencemag.org/lookup/doi/10.1126/sciadv.aao1378. [Internet].

[4] Carvalho AR, Pennino MG, Bellido JM, Olavo G. Small-scale shrimp fisheries bycatch: A multi-criteria approach for data-scarse situations. Marine Policy 2020;116:103613, Available from: https://linkinghub.elsevier.com/retrieve/pii/S0308597X18307164 . [Internet].

[5] Jiorle RP, Ahrens RNM, Allen MS. Assessing the utility of a smartphone app for recreational Fishery catch data. Fisheries 2016;41(12):758–66, Available from: http://doi.wiley.com/10.1080/03632415.2016.1249709 . [Internet].

[6] Gutowsky LFG, Gobin J, Burnett NJ, Chapman JM, Stoot LJ, Bliss S. Smartphones and digital tablets: Emerging tools for Fisheries professionals. Fisheries 2013;38(10):455–61, Available from: http://doi.wiley.com/10.1080/03632415.2013.838133. [Internet].

[7] Merrifield M, Gleason M, Bellquist L, Kauer K, Oberhoff D, Burt C, et al. eCatch: Enabling collaborative fisheries management with technology. Ecol Inf 2019;52:82–93, Available from: https://linkinghub.elsevier.com/retrieve/pii/S157495411830298X. [Internet].

[8] McInnes LC, Heroux MA, Draeger EW, Siegel A, Coghlan S, Antypas K. How community software ecosystems can unlock the potential of exascale computing. Nat Comput Sci 2021;1(2):92–4, Available from: https://www.nature.com/articles/s43588-021-00033-y . [Internet].

[9] Heron M, Hanson VL, Ricketts I. Open source and accessibility: Advantages and limitations. J Interaction Sci 2013;1(1):2. http://dx.doi.org/10.1186/2194-0827-1-2, Available from: [Internet].

[10] R Development Core Team R. R: The r project for statistical computing. 2016, Available from: https://www.r-project.org/.

[11] van Rossum G, Drake FL, Van Rossum G. The Python language reference. release 3.0.1 [repr.]. Hampton, NH: Python Software Foundation; 2010, Python documentation manual.

[12] Teh LSL, Teh LCL, Sumaila UR. Quantifying the overlooked socio-economic contribution of small-scale fisheries in Sabah, Malaysia. Fish Res 2011;110(3):450–8, Available from: https://linkinghub.elsevier.com/retrieve/pii/S0165783611002189. [Internet].

[13] de M.A. Damasio L, Lopes PFM, Guariento RD, Carvalho AR. Matching Fishers' knowledge and landing data to overcome data missing in small-scale Fisheries. In: Stergiou KI, editor. PLoS One 2015;10(7):e0133122, Available from: https://dx.plos.org/10.1371/journal.pone.0133122. [Internet].

[14] Eriksson H, Friedman K, Amos M, Bertram I, Pakoa K, Fisher R, et al. Geography limits island small-scale fishery production. Fish Fish 2018;19(2):308–20, Available from: https://onlinelibrary.wiley.com/doi/10.1111/faf.12255. [Internet].

[15] Mills DJ, Westlund L, de Graaf G, Kura Y, Willman R, Kelleher K. In: Pomeroy RS, Andrew NL, editors. Under-reported and undervalued: small-scale fisheries in the developing world. Wallingford: CABI; 2011, p. 1–5, Available from: http://www.cabi.org/cabebooks/ebook/20113041514.

[16] Bevilacqua AHV, Angelini R, Steenbeek J, Christensen V, Carvalho AR. Following the fish: The role of subsistence in a fish-based value chain. Ecol Econ 2019;159:326–34, Available from: https://linkinghub.elsevier.com/retrieve/pii/S0921800917309564. [Internet].

[17] Sulu RJ, Eriksson H, Schwarz A-M, Andrew NL, Orirana G, Sukulu M, et al. Livelihoods and Fisheries governance in a contemporary Pacific island setting. In: Ferse SCA, editor. PLoS One 2015;10(11):e0143516, Available from: https://dx.plos.org/10.1371/journal.pone.0143516 . [Internet].