A Lean Web Framework for Crowdsourcing

Ethan L. Nelson, University of Wisconsin–Madison

ethan.nelson@aos.wisc.edu

Motivations

- Scientists face an ever-growing quantity of data to grapple with in performing scientific studies.
- Oftentimes, analysis requires some aspect of qualitative human judgment, such as the original Dvorak technique (Dvorak 1975) and Fujita scale (Fujita 1971), or is dependent on subtle visible features like separating clouds and snowcover in a visible image.
- Crowdsourcing involves distributing small tasks to many users over the internet for completing manual tasks or receiving input from multiple people.
- Knapp et al. (2016) has utilized crowdsourcing in the "Cyclone Center" to extract characteristics of tropical cyclones using historical images.
- The Meteorological Phenomena Identification Near the Ground (mPING) project relies on citizen reports using a mobile phone application to identify weather information occurring at a location for dual polarization radar algorithm improvements (NSSL 2014).
- Issues of vendor lock-in or specific system dependence motivate the development of a platform-agnostic crowdsourcing system that can be self-hosted or hosted in the cloud.

Site Architecture

- Flask (Ronacher 2017) is a Python web library with a penchant towards lean web frameworks that loosely follows the model–viewer pattern.
- User information, image or other object links that are to be classified, and classification information is stored in a database backend; the use of SQLAlchemy (Bayer 2017) for database interface allows for flexibility in the database of choice (from SQLite to Postgresql to MySQL).
- Page layouts are generated using the Jinja2 (Ronacher 2014) templating engine.
- Either Flask (through an Apache virtual host) or an alternative web server (e.g. Apache, NGINX through a WSGI module) is employed to actually serve the application to a user.
- This means the application can be hosted on any machine, whether a local Unix box or "in the cloud" through resources like Heroku, Google App Engine, or Amazon Web Services.

User Workflow

- Currently, user registration is limited to a username, email, and password (Figure 1), but additional information can also be collected.
- After registering, users are presented with either the sole project (in the event only one project is created) or a list of projects they can select from.
- A tutorial can optionally be completed by the user before they are allowed to proceed with classification tasks.
- Users will be presented with one piece of information (be it an image or other object) and prompted to select an answer to the project's question.
- Though not yet implemented, user levels and achievements will be added based on the number of tasks a user completes.

Administration and Customization

- Branding is handled through a configuration file for page templates and styling files.
- Projects are created using a Python script with a list of images or other objects, a tutorial page, and the type of options to present to the user.
- An administration module to create projects through the web is also planned for the future.

Conclusions and Challenges

- This framework provides a mechanism for applying the power of crowdsourcing to any task, though the focus here is on meteorological and climatological tasks.
- Other crowdsourcing projects have ranged from manual transcription of old records, classification of historical images, and identification of features within data.
- The focus in this library is a lean web framework that requires minimal resources to host and similarly low requirements of user machines.
- A major drawback to self-hosting a crowdsourcing project is recruiting and engaging users.
- Projects employing this framework, as opposed to utilizing a platform like zooniverse.org, do face the challenge of user recruitment and engagement.
- Plugging into an existing social group is recommended when creating a new project or set of projects through this mechanism.

References

- Bayer, M., 2017: SQLALchemy. Accessed at `http://sqlalchemy.org`.
- Dvorak, V. F., 1975: Tropical cyclone intensity analysis and forecasting from satellite imagery. *MWR*, **103**, 420–430.
- Fujita, T. T., 1971: Proposed characterization of tornadoes and hurricanes by area and intensity. *Uni. of Chicago SMRP Res. Pap.*, **91**, 42pp.
- Knapp, K. R., J. L. Matthews, J. P. Kossin, and C. C. Hennon, 2016: Identification of tropical cyclone storm types using crowdsourcing. *MRW*, **144**, 3783–3798.
- National Severe Storms Laboratory, 2014: mPING Fact Sheet. Accessed at `http://www.nssl.noaa.gov/news/factsheets/mPING_Fact_Sheet_2014.pdf`.
- Ronacher, A., 2014: Jinja2. Accessed at `http://jinja.pocoo.org`.
- Ronacher, A., 2017: Flask. Accessed at `http://flask.pocoo.org`.

Acknowledgments

Thanks to NASA HQ under the Earth and Space Science Fellowship Program Grant #NNX14AL35H for providing base educational funding.

Code is available at `https://github.com/ethan-nelson/bartiromo` and a demo is hosted on `http://micro.ethan-nelson.com`.

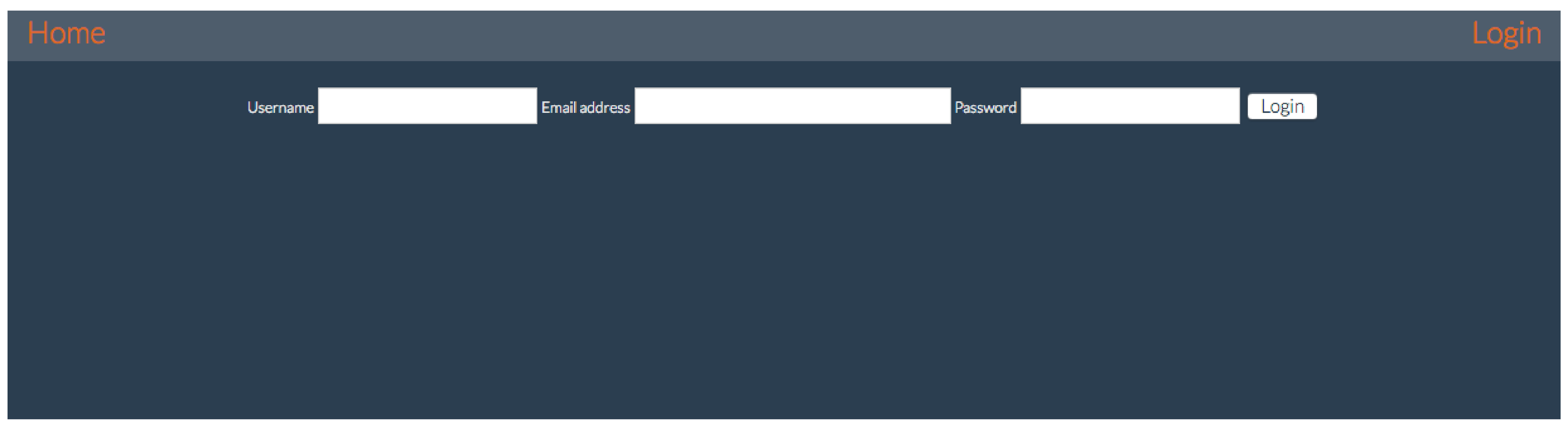

Figure 1: Sample registration screen with default style and requested user information.

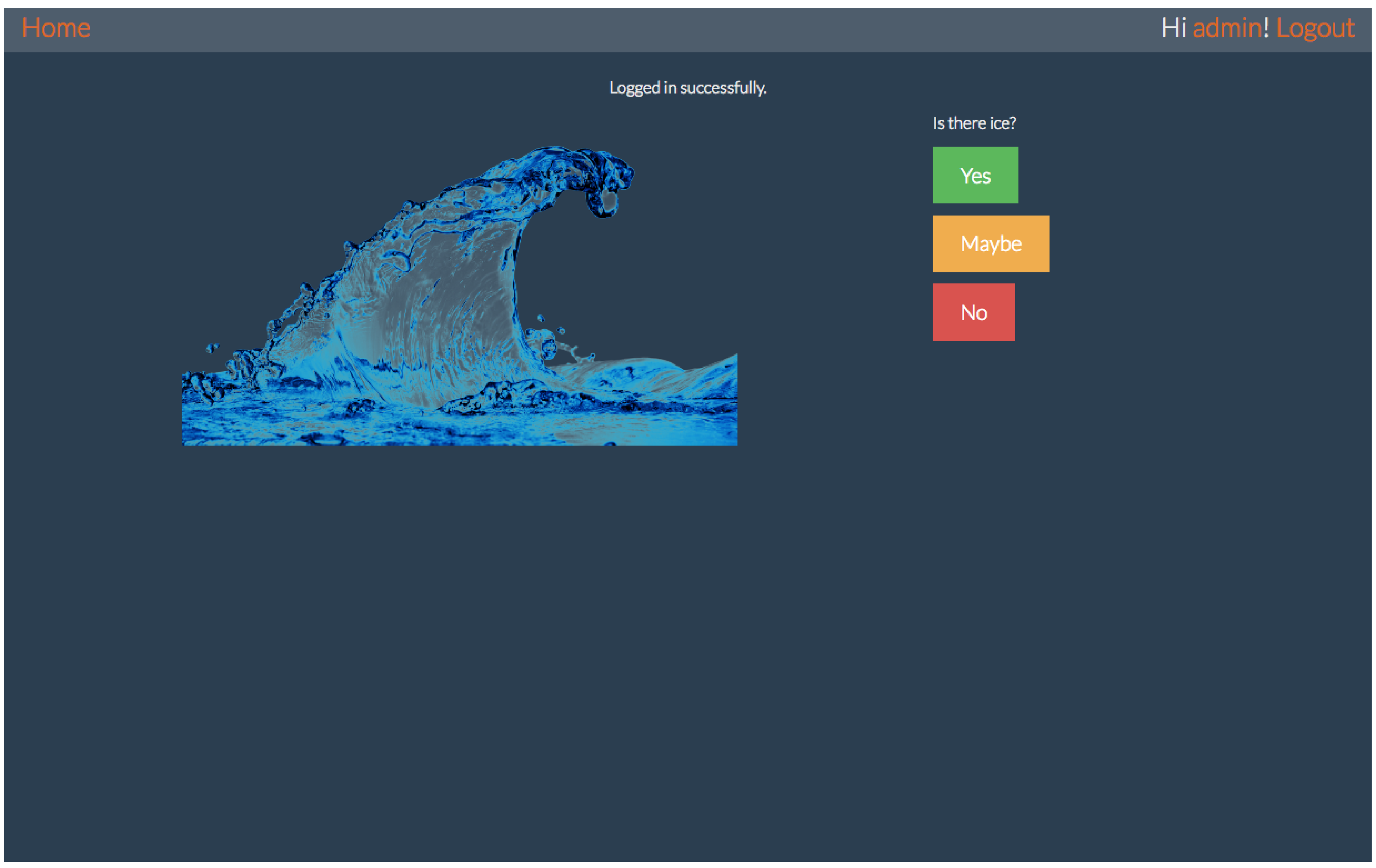

Figure 2: Sample yes-maybe-no classification for presence of ice within an image.