COLLEGE OF ENGINEERING

Electrical Engineering and Computer Science

SCIENCE SLEUTHS

Mobile and web application for teacher and student research collaboration

TEAM

Bruce Marandino
marandib@oregonstate.edu
Timothy Hong
hongti@oregonstate.edu
Patricia Booth
boothpat@oregonstate.edu

sleuths.herokuapp.com

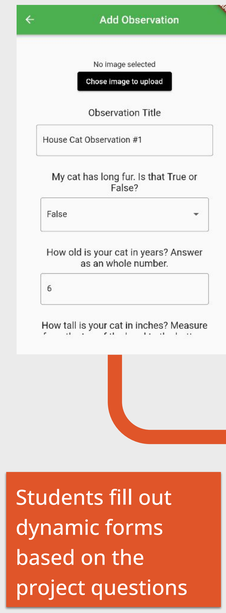
FEATURES

TEACHER WEB PORTAL — FLASK / REACT

- Initial user authentication through Google Firebase followed by JWT session authorization on backend
- Project management enables analysis of observation data submitted by displaying a table summary along with statistical graphs
- New projects can be created with a variable length of survey questions giving the option for a variety of data types including true or false, numeric, short answer, multiple choice, and date/time
- Project data can be exported as a .csv file allowing users to have an offline copy of observation results

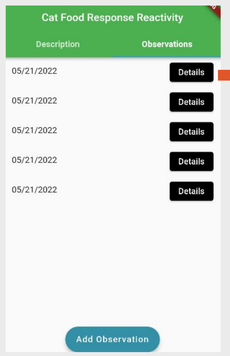
STUDENT MOBILE APP — FLUTTER

- App enables submission of project observation with data collected based on project parameters
- App remembers student information and current project while re-opening by utilizing user preferences
- Observation data is viewed in the form of a stream from Firestore allowing students to view updates in real time
- Students can upload images from their device's gallery alongside their observation posts
- Observations can be edited using a pre-filled form which remembers their original responses

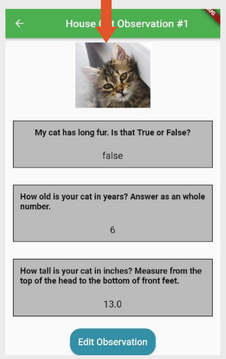


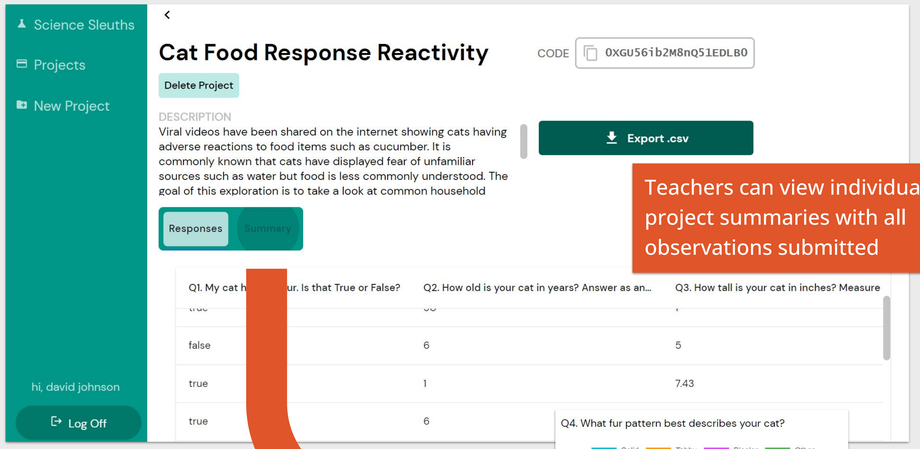
Students fill out dynamic forms based on the project questions

Observation list updates in real time and have detailed views

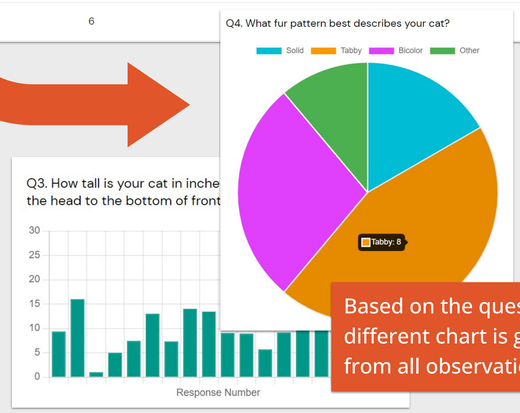


Detail view shows entire observation's information





Teachers can view individual project summaries with all observations submitted



Based on the question type, a different chart is generated from all observation data

CITIZEN SCIENCE APP

Citizen Science describes community-driven research where everyday citizens can contribute to science projects. Science Sleuths aims to be an educational platform that enables teachers and students to collaborate on scientific research. A cross-platform mobile application was developed in Flutter for students to submit observation data. This provides the advantage of allowing students to note observations anywhere they go with any mobile device they have. Collective observation data is displayed for teachers in the web portal built in Flask and React. In this portal, the teachers can manage existing projects, create new projects, and view detailed project summaries and relevant statistical information.

Our team was new to many of the technologies used in the project but we accepted the challenge. We learned that designing and developing two applications has the advantage of being independent entities for separations of concern but it also requires agreement on methods of communication used so both applications can send and receive the data seamlessly.

TECHNOLOGIES USED

- Google Firebase Authentication for teacher web portal authentication
- Google Firebase Firestore for NoSQL database
- Google Firebase Cloud Storage for student image storage
- React for rendering dynamic interfaces with JavaScript
- Material UI Library for displaying responsive components that adheres to material design guidelines
- Chart.js for displaying pie and bar graphs of observation results

```python
def create_project(project: "Project") -> str:
    """
    Takes a Project instance and adds it to Firestore db. Creates a Project
    Summary instance and adds it to the project's owner.
    :param project: the Project instance
    :return: the project_id
    """
    try:
        db = firestore.client()
        project_ref = db.collection(u'Projects').document()
        project_id = project_ref.id
        project_ref.set(project.to_dict())

        # create and add project summary to the project owner
        project_summary = ProjectSummary(project_id, project.get_title(),
                                         project.get_description())
        db.collection(u'Users').document(project.get_owner_id()) \
            .update({u'owned_projects': firestore
                    .ArrayUnion([project_summary.to_dict()])})

        return project_ref.id
    except exceptions.FirebaseError as e:
        print(e)
        return None
```

Python code in backend accessing Firestore database and adding a new project

Oregon State University