



9. Water for the World



Engineers Without Borders

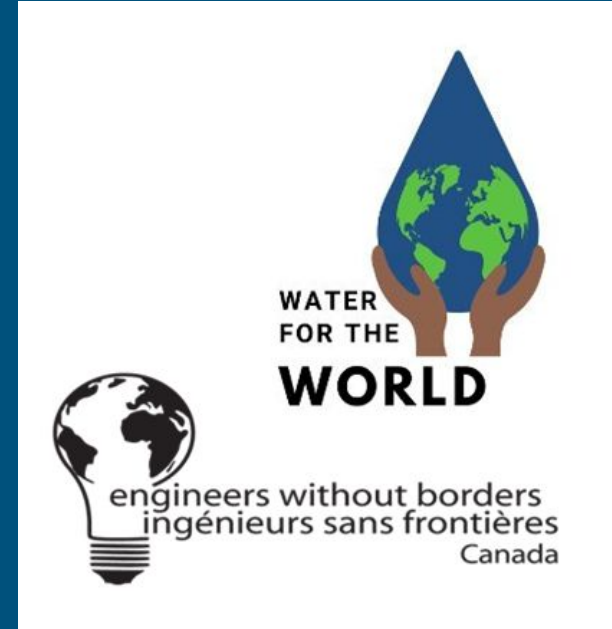


Git Good

1. Young Jun Cho
2. Glen Marco Jeremy
3. Justin Alianto
4. Nigel Yang
5. Sargunan Gurumurthy
6. Stefanus Albert Welong
7. Varun Pillai

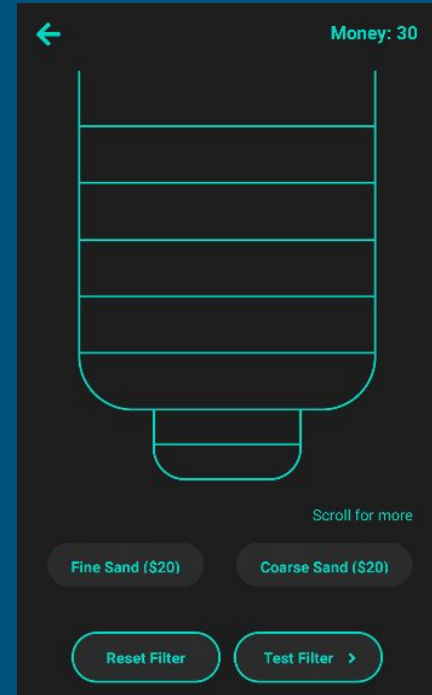
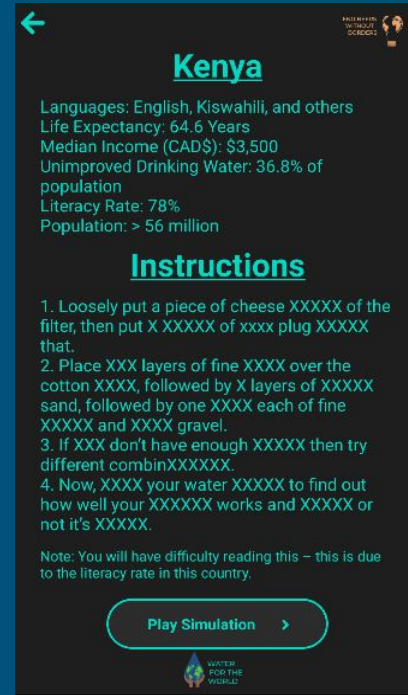
Partners and Problem

- ❖ Engineers Without Borders and Water for the World - WT4W
- ❖ Spread awareness (safe drinking water and literacy rate)
- ❖ Filter Building Workshop
- ❖ Covid and the transition to remote workshop



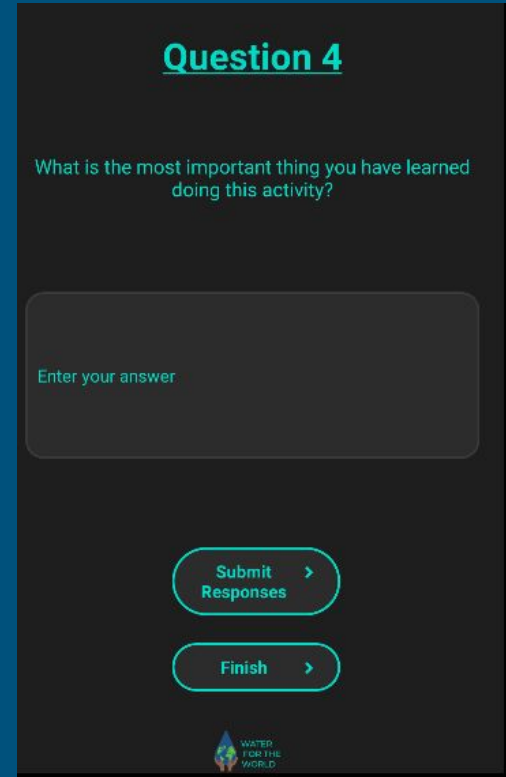
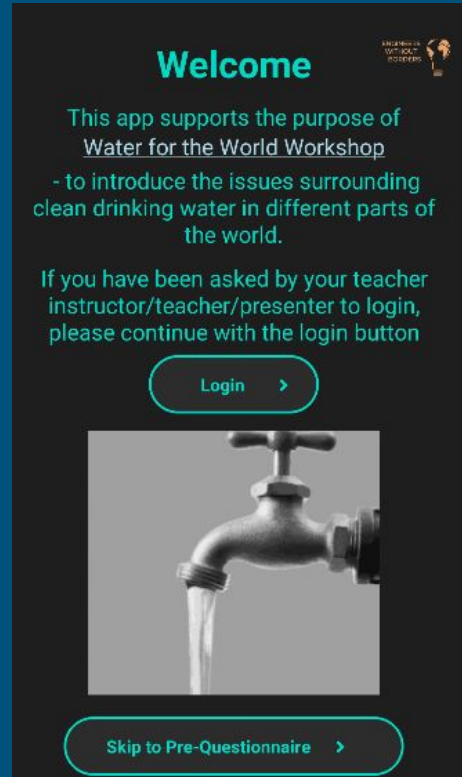
Audience and Purpose

- An educational tool for MS/HS students and teachers
- Anyone interested in learning
- Spread awareness about water safety and methods to clean drinking water
- Showcase disparities of global access to clean water
- Literacy rates and challenges



Different User Types

- Regular user
 - Single player
- Student
 - Join Homeroom
- Teacher
 - Create Homeroom



Demo!

Key Features:

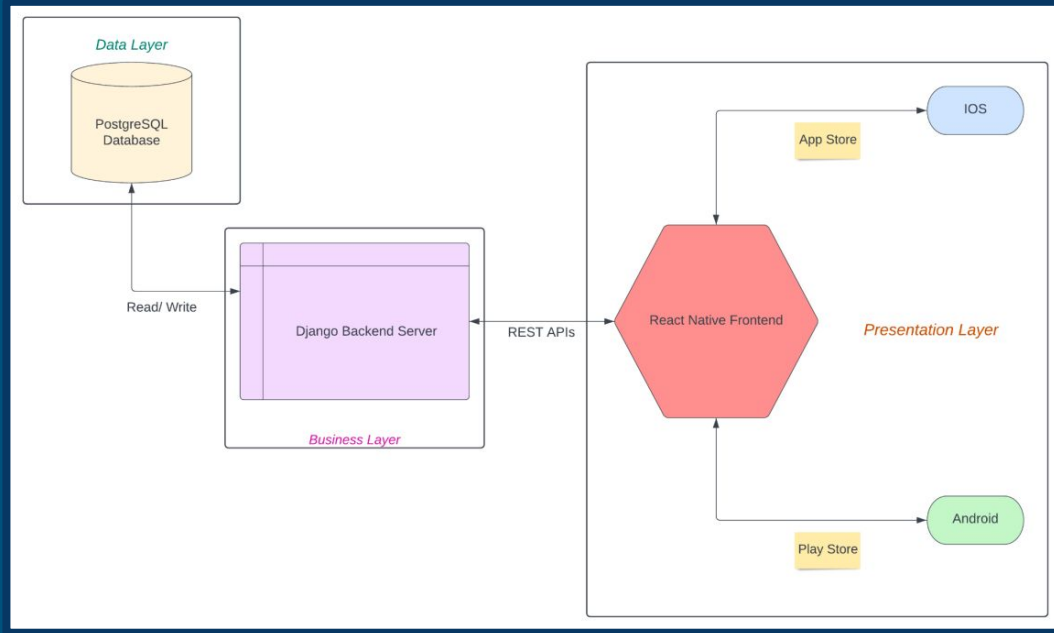
- Links to external sources
- Login (for educational purposes) vs skip login (for anyone)
- Pre-questionnaires (includes sliders and buttons)
- Filter building (different countries, literacy rates, money)
- Post-questionnaires (ability to send email to teacher)

Software Tools

Why PostgreSQL, Django, and React Native?

- PostgreSQL scalability
- Ease of integration and use between Django and PostgreSQL
- Ease of integration between Django REST framework and React Native using REST APIs
- React Native uses JSON as its primary data exchange format, which PostgreSQL has excellent support for
- React Native has cross-platform compatibility for deployment (iOS, Android)
- Alignment with the team's level of expertise and goals

Software Architecture

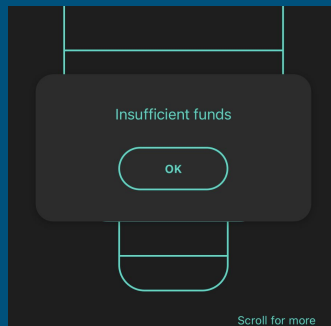


Coding Practices & Documentation

- Structured Repository
- Descriptive Names and Messages
- Python (Django) & JS (React Native) Conventions

```
class CreateRoomView(APIView):
```

```
// Images of the filter materials
const fineSandImg = require('../assets/fineSand.jpeg');
const coarseSandImg = require('../assets/coarseSand.jpeg');
const fineGravelImg = require('../assets/fineGravel.jpeg');
const coarseGravelImg = require('../assets/coarseGravel.jpeg');
const cottonImg = require('../assets/cotton.jpeg');
const cheeseClothImg = require('../assets/cheeseCloth.jpeg');
```



```
▼ backend
  > accounts
  > backend
  > homeroom
  > quiz
  .gitignore
  manage.py
  Procfile
  requirements.txt
▼ frontend
  > .expo
  > apis
  > assets
  > components
```

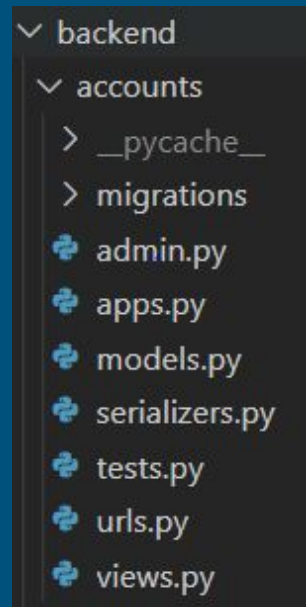
Coding Practices & Documentation

- Modular Code & DRY (Don't Repeat Yourself)
- Automated Testings (APIs, Database, etc)
- Proper Documentations (Comments, Docstrings, README.md)

```
# API Endpoints Tests
class AnswerViewTestCase(APITestCase):
# Database Tests
class AnswerModelTestCase(TestCase):
```

```
class CreateRoomView(APIView):
    permission_classes = [IsAuthenticated]

    def post(self, request):
        """Creates a new homeroom and assign the teacher who created it
        as the teacher of the room."""
```



Process & Deployment



- Deployment Tools
 - Database server uses Railway.app
 - Backend server deployed on Heroku, which can run the server 24/7
 - Front end utilized Expo go: a free, open-source client for testing React Native apps
- Where and How Our Product is Deployed
 - .apk app using EAS (Expo Application Services)
- Team Workflow
 - Gitflow Workflow - develop 'd2-test' branch, release branch
 - Pull requests for new features
 - Scrum meetings
 - GitHub issues
 - DevOps

How to Access the Application

Communication with our partner:

- Regular emails, Zoom meetings

The .apk file is located here:

<https://drive.google.com/file/d/1LJXTgxYDAb0gpsShYzucemnNIBzSUJGO/view?usp=sharing>

Future Plans:

- Google Play Store
- iOS App Store
- .apk On EWB and W4TW Website

Handing-off our Application:

- Open-source license
- Architectural rules

Reflection & Key Learnings

What did we learn?

- The benefits of agile methodology to fasten the development process.
- Understanding the project requirements is crucial for project success.

What would we do differently next time?

- Ensure a clear understanding of project requirements.
- Approach subteam division more strategically.

Future Plans

Features:

- Multiplayer version of the game filter
 - Sharing of funds between countries
 - Random generator of events affecting countries' resources

Others:

- Standardize app design throughout devices
- More comments and documentations
- Frontend Testings
- Deployment on iOS, Android, and EWB and W4TW websites

Individual Contributions

1. Young Jun Cho: Backend, Frontend
2. Glen Marco Jeremy: Database, QA analysis, app design
3. Justin Alianto: Database, QA analysis, app design & layout
4. Nigel Yang: Back-end, Front-end, Deployment
5. Sargunan Gurumurthy: Front-end, app design & layout
6. Stefanus Albert Welong: Front-end, app design & layout
7. Varun Pillai: Back-end, Front-end



Thank You!

