Koobecaf

Connecting People

The Team

Rushikesh Pharate

Full Stack Developer

Mohit Alumullithody

Product Manager

Tony Dattolo

Full Stack Developer

Rahul Shamdasani

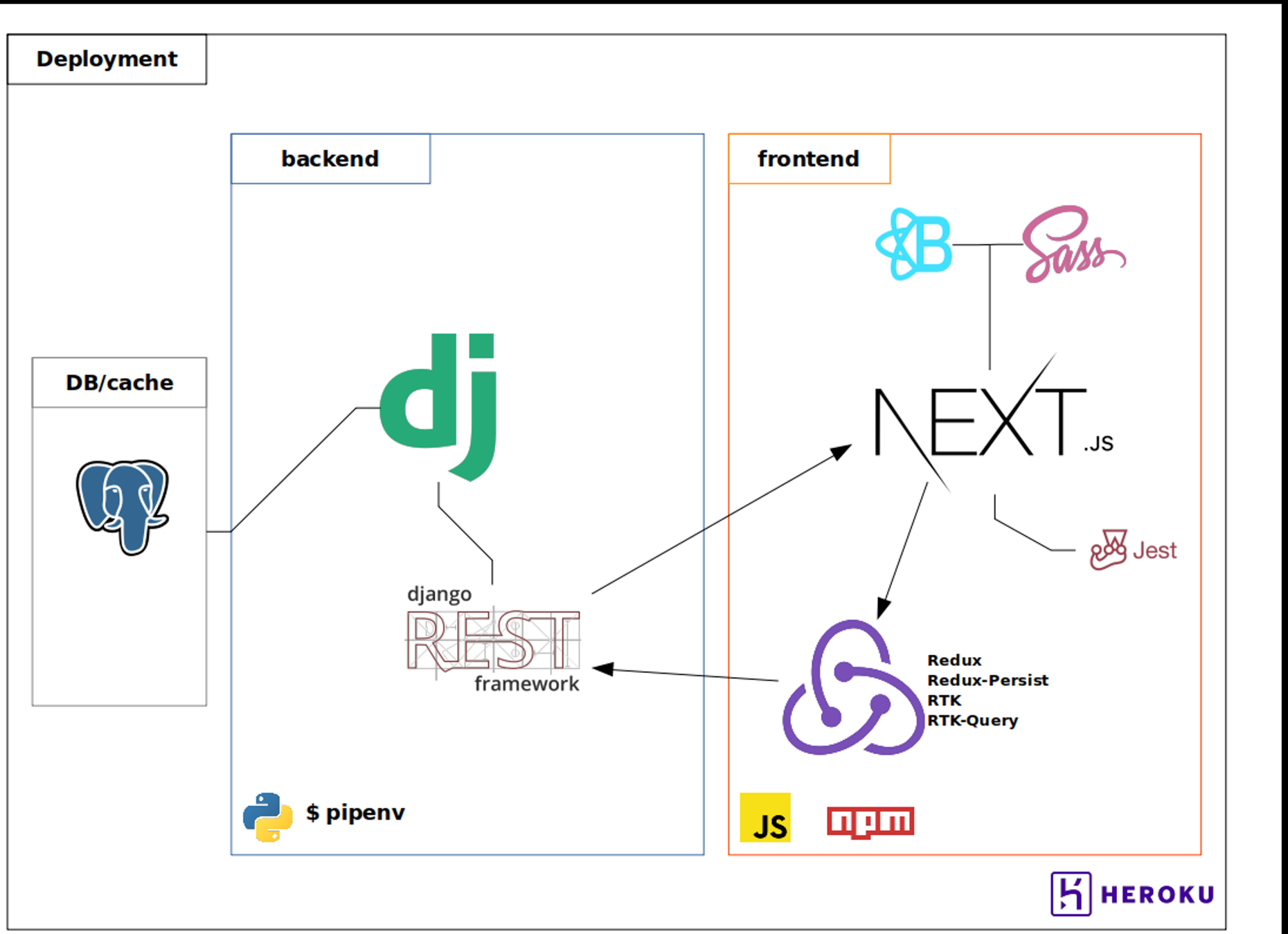
DevOps/Frontend Developer

Completed Features

yayyy

- Admin panel
- API endpoint GUI browser for devs
- User Registration, Activation, Resend Activation, Reset Password, Confirm reset
- User Login and Authentication with JWTs
- Profile, Edit Profile, display posts by user on profile, allow delete posts on profile page
- Feed displaying all posts in descending order
- Explore users and todays posts
- Direct message other users
- Create a fan page, with followers
- Create events
- Recieve notifications for events
- Client Satisfied ? -> **YES**

Tech Stack


Deployment
backend
frontend
DB/cache
dj
Sass
NEXT.JS
Jest
django
REST
framework
Redux
Redux-Persist
RTK
RTK-Query
$ pipenv
JS
npm
HEROKU

Not Completed Features

booooo

- Google Auth. It’s working but there’s a known open bug with NextJS and Django
- Marketplace (Optional)
- Images/videos - Time constraints

Known Bugs

booooo x2

- Default profile name needs to be randomized like reddit
- date on posts being miscalculated in prod
- send message from user profile doesn’t autofill sender/receiver values
- JWT refresh logic buggy

Future Iterations

the future

- Responsive UI
- Dockerization
- UX/UI improvements, affordances
- CDN hosting for images/videos
- Workaround for google auth
- Automated QA CI/CD integration

Tools

stuff we used to do stuff

github

vscode

whatsapp

jira

teams

zoom

heroku

Product Timeline

	Week										
	1	2	3	4	5	6	7	8	9	10	11
Sprint 1	Initial Setup & Login										
Sprint 2			OAuth & Profile Page								
Sprint 3					Posts & Hosting CI/CD						
Sprint 4							Search & Dark Mode				
Sprint 5									Pages, Events, Messages, Explore & Notifications		

Team Member Contribution

Member	Responsibilities
Tony Dattolo	User Registration, Auth, Profile, Posts, Explore, Messages
Rushikesh Pharate	Posts, Pages, Events, Notifications
Rahul Shamdasani	Profile, UI Improvement, Deployment
Mohit Alumullithody	Search, Jira Management, Documentation, User Registration, QA

Design Patterns

reusability

Factory Pattern: RTK-Query Hooks

Model View Template: Rest Framework

Finite State Machine: Redux Toolkit

Persistence Pattern: Redux Persist

Separation of Concerns

decoupled, modular

Data model: models.py

Business logic: views.py

UI: react components

styling: scss modules

state management: RTK slices

query logic: rtk-q createApi mutations

Development Workflow

16 steps to glory

1. Open the project in VSCode
2. Create a new feature-branch off of the develop branch
3. Open terminal and activate virtual environment inside of the project: pipenv shell
4. Run: python manage.py startapp <appname>
5. Register the new app in our base settings.py, e.g.: ‘contracts.apps.ContractsConfig’
6. Add the api endpoints in our base *project* level urls.py, including the *app* level urls.py
7. Create new model from wanted data model attributes for that feature in models.py
8. Create serializers for each model in serializers.py
9. Create CRUD or other views in views.py
10. Register the endpoints for the API layer in urls.py, calling upon the created views
11. Create new RTK/RTK-Q slice/api files for that specific feature
12. Register the RTK slice and RTK-Q api inside of our persisted redux store.js
13. Define state changes to the data in the slice if needed
14. Define http request queries in the featureAPI.js file, declare and export hooks through destructuring
15. Create component, call hooks to make requests, and handle the common states associated with them
16. Style the component as needed inside of its respective scss module file, or inline for smaller things

Koobecaf

Accessible @ https://koobecaffrontend.herokuapp.com/

the end

thank you, and time permitting we'll do Q&A