Front-end Test

Summary

Write a simple messaging app that allows users to send and receive messages. This app must use the provided API for it's core functionality. You must implement all of the below routes as basic UI functionality. Advanced styling is unnecessary but the app must be functional and intuitive enough to use without guidance.

API Details

API URL: https://messaging-test.bixly.com
Auth Scheme: Token
Paradigm: RESTful

API Resource Map:

messages/ GET - List Messages
messages/ POST - Create Message
```
{
        "title": "<title:string>",
        "body": "<password:string>",
        "receiver": "<receiver_username:string>"
}
```
messages/sent/ GET - List Sent Messages
messages/sent/ DELETE - Delete Sent Message
messages/<message_id>/ GET - Get Message Detail
messages/<message_id>/ DELETE - Delete Message
api-token-auth/ POST - Exchange credentials for auth token
```
{
        "username": "<username:string>",
        "password": "<password:string>"
}
```
NOTE: You MUST format the token in your header exactly as it is below
`Authorization: Token <token>`

Test user:

Username: test
Password: test123!
USE THIS TO LOG IN VIA THE ABOVE TOKEN AUTH ENDPOINT

Test receiver user:

Username: tomk

The API accepts most common forms of data payload, but defaults to `x-www-form-urlencoded` unless otherwise specified within the `Content-Type` header. For JSON payloads you must set `Content-Type: application/json`.

Requirements
The app should include the following features:
- Login screen (should be able to save authentication token to skip login after initial login)
- Inbox (list of messages received)
- Sent (list of messages sent)
- Delete (both Inbox and Sent screens should allow users to delete message)
- Compose (includes recipient, title, and body)
- Error messages (should show an error if any of the requests are invalid)

Please include a brief README file that includes any instructions we need to set up your project, any necessary external packages, and anything else that would affect getting your project to run.

When complete, please ZIP up the project and send it via email or google drive to your test administrator by the assigned deadline.