



# HRDC EMPLOYEE INTRANET

## DEVELOPER INFO & RESOURCES





# SOURCE CODE

## Obtaining Source Code

1. Visit <https://github.com/423S25/repo4>
2. Clone the repository to your local machine  
Git clone <your chosen repo name>

## How to Build

- Visit this documentation: <https://nextjs.org/docs/app/getting-started/installation>
- Install NextJS
- Visit this documentation: <https://flask.palletsprojects.com/en/stable/installation/>
- Install these npm modules
- npm install lucide-react
- npm install next

Gain access as to the GitHub as a contributor

Speak to your superiors to be added

## FRAMEWORK DOCUMENTATION

NEXT.JS documentation	<a href="https://nextjs.org/docs">https://nextjs.org/docs</a>
Tailwind CSS documentation	<a href="https://v2.tailwindcss.com/docs/installation">https://v2.tailwindcss.com/docs/installation</a>



# STRUCTURE

Directory Structure:

## Documentation

- Developer Doc
- User Doc
- UML
  - UML documents

## Site

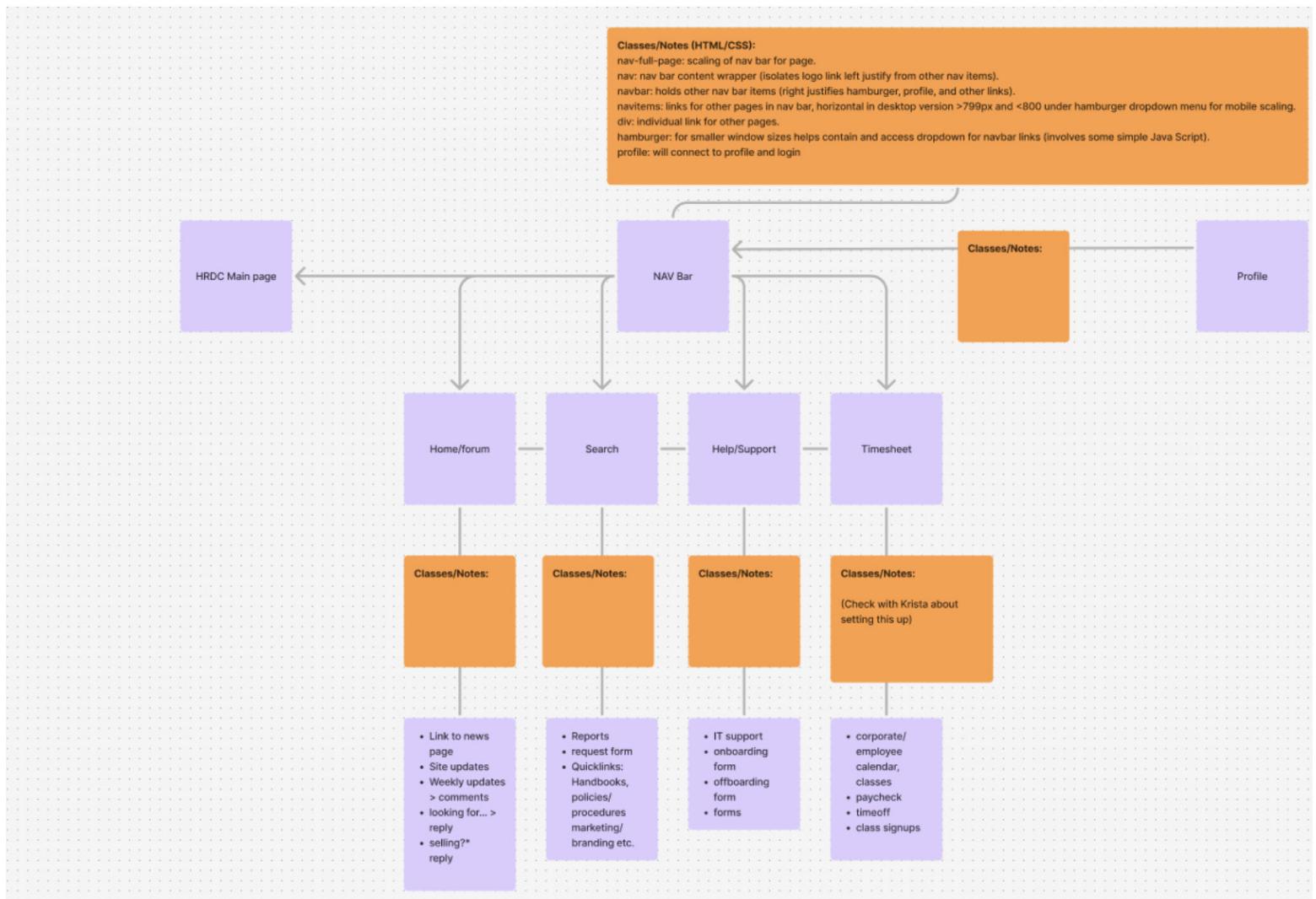
- Design
  - Images
  - Styles
  - Frameworks

## docs

- NextJS
- HTML
- Misc. Code



# CLASS DIAGRAM





# TESTING

How to Test:

DO NOT push to Github before testing

After saving your changes, access the website <https://hrdc.fly.dev/> via flask

Test your feature to ensure it doesn't negatively impact other areas.

Have team members review your changes.

Push afterwards

Automated Build and Test:

1. Access project repository
2. Run automated test suite
  - Available through Github Actions tab
3. View passed/failed tests
  - Failed tests have error terminals that you can view

Debugging and Issues

Current Bugs will be tracked through Github Issues



# TESTING PROTOCOL AND METHODS

## Cross-Browser Testing

- Ensure compatibility and functionality across various browsers and devices.
- Manual testing on: Firefox (Windows), Edge (Windows), Chrome (Windows), iPhone
- Chrome (Mobile)
- Verify layout, functionality, and responsiveness across different browsers and devices.

## Integration Testing

- Validate the integration of new features and changes with existing codebase.
- Feature branching: Create separate branches for new features or changes.
- Testing before committing to main: Thoroughly test new features or changes before merging with the main branch.
- Manual testing of integrated features to ensure seamless functionality.

## Black-Box Testing

- Identify issues and bugs through exploratory testing.
- Manually explore the website, attempting to break or identify issues.
- Verification of expected behavior and identification of unexpected issues.

## Automated Testing with GitHub

- Automate testing and validation of code changes.
- Utilize GitHub Actions for automated testing.
- Create a .yaml file to define automated tests and workflows.
- Integration with GitHub Issues for tracking and resolving identified issues.



# TICKETING EMAIL CHANGES

## Changing the Support Ticket Email Configuration

This section explains how to change the email address where support tickets are sent from a Gmail account to your custom domain (@thehrdc.org).

### Current Configuration

The ticketing system is currently configured to use Gmail for sending support tickets. The system uses the following environment variables:

EMAIL\_USER: The email username/address  
EMAIL\_PASS: The email password or app password  
EMAIL\_HOST: The email server host (currently set to Gmail's SMTP server)  
EMAIL\_PORT: The port used for email connections  
EMAIL\_SECURE: Boolean to indicate if secure connection is required  
GOOGLE\_API\_KEY: API key for Google services

### Steps to Change Email Configuration

#### 1. Update Environment Variables

Modify your .env file with the following changes:

```
# Change from Gmail to your domain
EMAIL_USER=support@thehrdc.org          # Replace with your preferred email address
EMAIL_PASS=your_new_password              # Set the password for your domain email
EMAIL_HOST=mail.thehrdc.org                # Replace with your domain's mail server
EMAIL_PORT=587                            # Common port for SMTP (confirm with your email provider)
EMAIL_SECURE=true                         # Set to true if using SSL/TLS
```

If using a different email provider than your current domain host, you may need additional



# TICKETING EMAIL CHANGES

configuration. Consult with your email provider if this is the case.

## 2. Testing the Configuration

After updating the environment variables:

Restart the server

Submit a test ticket through the form

Verify that the email is received at the new address

Check that confirmation/failed messages appear after ticket submission

## 4. Troubleshooting Common Issues

Authentication errors: Verify your username and password are correct

Connection timeout: Check that EMAIL\_HOST and EMAIL\_PORT are correct

SSL/TLS issues: Adjust EMAIL\_SECURE as needed

Firewall issues: Ensure your server can connect to the email provider

## Note

Keep the GOOGLE\_API\_KEY variable even if you are not using gmail, it is necessary for the functionality of the search, calendar, and drive features.



# ADDING FILES TO GOOGLE DRIVE

Search Feature – Adding Files

Find (or create) your target folder in Drive

Go to <https://drive.google.com/> and sign in with the Google account that owns your folder.

Navigate to the folder you want your app to search.

In your browser's address bar you'll see a URL like:  
<https://drive.google.com/drive/folders/1Kf4QwSSCj03NeD4L38RtML3HP67xYcu8>

Copy everything after /folders/ — that's your folder ID.

Tell your app about that folder

In your .env.local, add the following, changing the ID to match yours:  
GOOGLE\_DRIVE\_FOLDER\_ID=1Kf4QwSSCj03NeD4L38RtML3HP67xYcu8

Then restart your dev server: npm run dev

Share the folder with your service account

Because your API calls authenticate as a service account, that account needs permission on the folder:

In Drive, right-click the folder and choose Share.

Paste your service account email (for example:  
hrdc-745@advance-copilot-457622-a0.iam.gserviceaccount.com)

Grant it Viewer access.



# ADDING FILES TO GOOGLE DRIVE

Upload files into that folder

Drag & drop files onto the folder in the Drive UI, or click New → File upload.

When the upload finishes, your Next.js API call will include them.

Verify in your app

Open your search page and click the “ ” button (or submit an empty search) to run `loadAllFiles()`.

You should now see every file you added.



# FEATURE RELEASE

How to Release a Version:

Consult your PM

After consulting, speak with your clients about the changes

Learn what they are satisfied/dissatisfied with

Update documentation to reflect changes

Push new version to Github

(Optionally) Create blog release overviewing new features



# FURTHER QUESTIONS AND SUPPORT

## TROUBLESHOOTING AND SUPPORT

For internal assistance, contact the support team at [HRDC IT team]

For Developer support, contact loganschuman406@gmail.com.

## GOOGLE DRIVE HELP AND SUPPORT

<https://support.google.com/drive/?hl=en#topic=14940>

## FRAMEWORK DOCUMENTATION/SUPPORT

NEXT.JS documentation <https://nextjs.org/docs>

Tailwind CSS documentation <https://v2.tailwindcss.com/docs>