

# eCFR Assessment Setup

## Instructions for Testing the eCFR Assessment Project Locally

### Prerequisites:

1. Ensure you have `Node.js` and `npm` installed on your machine.
2. Make sure you have `MySQL` installed and running.

### Setting Up the Backend:

1. **Unzip the Project:**
  - Extract the project files to your desired location.
2. **Configure Environment Variables:**
  - Create a `.env` file in the `backend` directory.
  - Define your database credentials ( `DB_HOST` , `DB_USER` , `DB_PASSWORD` , `DB_NAME` ).
3. **Initialize the Database:**
  - Use the `database/schema.sql` file to set up your MySQL database.
  - Run the SQL script to create necessary tables.
4. **Install Backend Dependencies:**

```
cd /path/to/project/backend  
npm install
```

4. **Start the Backend Server:**

```
npm start
```

- The server will run on `http://localhost:3000` by default (check for changes in `index.js` if different).

### Setting Up the Frontend:

1. **Install Frontend Dependencies:**

```
cd /path/to/project/frontend  
npm install
```

1. **Start the Frontend Server:**

```
npm start
```

- This will open the frontend in your default browser at `http://localhost:3000`.

## Testing the Application:

### 1. Verify Backend is Running:

- Make sure there are no errors in the terminal output.
- Use a tool like `Postman` to test a couple of endpoints (e.g., `/agencies`).

### 2. Verify Frontend is Running:

- Ensure the UI loads without issues.
- Navigate through the different views: Download, Agencies, Word Count, etc.

### 3. Use the Application:

- Test the data download functionality via the UI.
- Check data rendering on the Agencies feed and analysis pages.

## Potential Issues:

- **CORS Issues:** Confirm that the `ngrok-skip-browser-warning` header is correctly set in frontend requests.
- **Network Issues:** Ensure ngrok is running if using a tunnel for backend communication.
- **Database Connection:** Verify database credentials in `.env`.

## Reviewer Notes:

- You can view the logic in `backend/DA0/EcrfDA0.js` for insight into database interactions and in `frontend/src` for UI logic.
- Look into the complexity calculation in the backend for evaluation of the regulatory burden.