CSE 340 Approval Final Project

This final project is copyrighted and may not be copied, duplicated, shared or posted online in any format! The solution, written by you, is a derivative work of this project and is also copyrighted. It may not be shared, posted, or made available in any form, other than to your GitHub repository and production server for the purposes of this class.

This project is intended to stretch you and be a thorough review of all concepts introduced in the course. You are encouraged to work with your learning team to finish the project, but all work must be your own.

Overview

During previous work, you built the ability to add new classifications and inventory items to the database, with these items becoming immediately available. While convenient, this process is not recommended due to the opportunity for error and possible embarrassment to the business. This final project adds safeguards to prevent such errors.

Tasks

1. Build into the existing inventory system a means that will prevent newly added classifications and inventory items from being immediately available on the website. Additionally, do not show classifications in the navigation menu if they do not have any approved inventory items.
2. Build a tool that is available only to administrators to show new classifications and inventory items that have been added to the site, but not yet approved. This should be available from the account management view.
3. Build a tool that allows administrators to approve or reject new classifications and/or inventory items. If rejected, the item is deleted from the database. If approved, the item becomes live on the site. Inventory items cannot be added to an unapproved classification.
4. When approved, the administrator's id is added to the database showing that it was them that approved the item and the date the approval happened.

Testing

Thoroughly test all aspects of the completed processes both locally and in the production environment. Use the existing accounts from the Unit 5, JWT Authorization activity for all inserts and approvals.

Submission

1. Zip the working code, and submit to the final project assignment.
2. Supply the production URL from your site as an assignment comment.
3. Submission must be by Saturday, 11:59pm of Week 13. Late work will not be accepted.

Grading Matrix

Obj. 1

- All views must meet the requirements of the frontend checklist, be professional and usable.

Obj. 2

- All processes must be fully operational as described in the task list.

Obj. 3

- All processes must use an MVC approach and be fully operational.

Obj. 4

- All model processes must use prepared statements and be fully operational.
- SQL revisions to the database must be stored in the "database" folder of the project.

Obj. 5

- Middleware must be used to limit access to the correct client and protect the database.

Obj. 6

- The submission is on-time and correct.

SQL Code

```sql
-- Table: public.classification

ALTER TABLE public.classification
    ADD COLUMN IF NOT EXISTS classification_approved boolean DEFAULT false,
    ADD COLUMN IF NOT EXISTS account_id integer,
    ADD COLUMN IF NOT EXISTS classification_approval_date timestamp DEFAULT CURRENT_TIMESTAMP,
    ADD CONSTRAINT account_id FOREIGN KEY (account_id)
        REFERENCES public.account (account_id) MATCH SIMPLE
        ON UPDATE NO ACTION
        ON DELETE NO ACTION
        NOT VALID;

CREATE INDEX IF NOT EXISTS fki_account_id
    ON public.classification USING btree
    (account_id ASC NULLS LAST)
    TABLESPACE pg_default;

-- Table: public.inventory

ALTER TABLE public.inventory
    ADD COLUMN IF NOT EXISTS inv_approved boolean DEFAULT false,
    ADD COLUMN IF NOT EXISTS account_id integer,
    ADD COLUMN IF NOT EXISTS inv_approved_date timestamp DEFAULT CURRENT_TIMESTAMP,
    ADD CONSTRAINT account_id FOREIGN KEY (account_id)
        REFERENCES public.account (account_id) MATCH SIMPLE
        ON UPDATE NO ACTION
        ON DELETE NO ACTION
        NOT VALID;

CREATE INDEX IF NOT EXISTS fki_account_id
    ON public.inventory USING btree
    (account_id ASC NULLS LAST)
    TABLESPACE pg_default;
```