Requirements:

1. Manual Verification Only
   All lost item reports, found item reports, and claim requests must be manually verified by an admin. No automation should make decisions.

2. Admin Queues
   Admin must have the following queues:
   - Pending Found Item Reports
   - Pending Lost Item Reports (optional)
   - Pending Claims

3. Found Item Verification Flow
   - Students can submit a Found Item Form.
   - Submitted found items go to the Pending Found queue.
   - Admin can: Approve, Reject, or Request More Information.
   - When admin approves, the item is published to the dashboard but with blurred or partially hidden descriptions/images.
   - Only admins can view the full, unblurred details.

4. Claim Request Flow
   - Students can request to claim a found item.
   - They must fill out a claim form with proof
   - Claim requests go to the Pending Claims queue.
   - Admin manually verifies the claim by comparing the proof with the unblurred found item details.
   - Admin actions for claims:
     - Approve Claim (item becomes "Ready for Release")
     - Deny Claim (with reason)
     - Request More Proof
     - Hold Item for X Days

5. Hold Item for X Days
   - Admin must have an option to place an item on hold for a specified number of days.
   - When on hold, the item cannot be released.
   - Admin must be able to set the number of hold days in Settings.

6. Lost Item Flow
   - Students can submit Lost Item Reports.
   - Admin verifies each submission.
   - If the lost item matches an existing found item, admin notifies the owner.
   - If no match exists, the item is posted as "Lost" until a matching found item is reported.

7. Item Lifecycle

Each item should have a full timeline:
- Reported
- Verified
- Published
- Claimed / Denied / On Hold
- Archived or Disposed (after retention period)

8. Admin Features
- View full unblurred item details
- View reporter or claimer information
- Add admin notes for every decision
- Audit log for all admin actions
- Filters for date, status, category, location
- CSV export for reports
- Settings page for:
  - Hold period days
  - Publish blur level
  - Admin contact info
  - Notification templates

9. Notifications
Include in-app and email notifications for:
- Found item approved
- Claim request submitted
- Claim approved
- Claim denied
- Item ready for claiming
- Lost item matched with a found item

10. UI Components
Include basic admin UI:
- Dashboard with stats
- Pending Found Items page
- Review Found Item Modal
- Pending Claims page
- Claim Review Modal with side-by-side comparison
- Settings page
- Item Timeline View

11. Output Format
Deliver:
- Full project folder structure
- Backend code
- Admin frontend pages

- Tailwind components
- Database migrations
- Sample environment file
- Sample seed data (lost items, found items, claims)
- README with setup instructions

Registration Logic:

- Students can register themselves through the normal signup route → role defaults to ‘student’.
- Admin accounts CANNOT be created through the public signup route.
- Admin accounts must be created manually (seed data, protected /admin/create route, or manual database insert).
- Prevent users from assigning themselves an admin role.

Login Logic:

- On login, verify email + password.
- After successful login, return:
    - JWT token
    - user id
    - user role
    - user name

Role-Based Access Control:

- Students:
    - can report lost items
    - can report found items

    - can request to claim an item
    - can search the lost & found dashboard
- Admin:
    - can view all pending lost/found posts for verification
    - can approve or reject lost/found posts
    - can approve or reject claim requests
    - can manage users
    - can view logs or history
    - can update item status (claimed, awaiting verification, etc.)

Middleware:

- Require a middleware that checks JWT, extracts user role, and blocks unauthorized actions.
- Example middlewares:
    - authRequired (checks if user is logged in)
    - adminOnly (only admin can access route)
    - studentOnly (only students can access student actions)

Behavior:

- When a user logs in, the frontend will check:
    - If role == 'admin' → redirect to admin dashboard
    - If role == 'student' → redirect to student dashboard

Security:

- Hash passwords using bcrypt.
- Store JWT secret in environment variables.
- Do not send passwords back in responses.”**