B12-A11_category-06

Public Infrastructure Issue Reporting System

Explanation video: ***B12_A11_category_06.mp4***

Description

The **Public Infrastructure Issue Reporting System** is a digital platform that enables citizens to report real-world public issues such as broken streetlights, potholes, water leakage, garbage overflow, damaged footpaths, etc. Government staff and admins can manage, verify, assign, and resolve reported issues efficiently.

Municipal services often suffer from delayed response and lack of tracking. Citizens have no centralized platform to report problems. This system:

- Improves transparency
- Reduces response time
- Helps collect and analyze infrastructure data
- Makes city service delivery more efficient

How the System Works

1. Citizens submit a report with issue details, photos, and location.
2. Admin reviews & assigns the issue to staff.
3. Staff verifies the issue and updates progress.
4. System tracks status from `Pending → In-Progress → Resolved → Closed`.
5. Citizens get updates and can track their issue anytime.
6. Premium citizens get priority support.

You are expected to deliver a **polished, bug-free, and impressive application** that stands out in terms of both functionality and design.

Good luck – build something amazing!

Main Requirements (Must Do)

- At least **20 meaningful GitHub commits** on the **client-side** code
- At least **12 meaningful GitHub commits** on the **server-side** code
- Create a good readme.md file that includes:
  – Website name
  – Admin email & password
  – Live site URL
  – At least 10 bullet points about your website features
- The website must be **fully responsive (mobile, tablet, desktop)**. The dashboard should also be responsive.
- Private routes must stay logged in after page refresh (no redirect to login).
- Hide Firebase and MongoDB secrets using environment variables.
- No Lorem Ipsum text anywhere.
- Use **sweet alert / toast / notification *(not browser alert)*** for login, signup, and all CRUD actions.
- Use **TanStack Query** for all data fetching.

HOME PAGE

1. Navbar
   - Logo + Website Name
   - Menu: Home, All Issues, 2 Extra page
   - When Logged in → show user profile picture
   - Click profile picture → dropdown with: User Name, Dashboard, Logout
2. Banner Section
   - Beautiful big banner/slider
   - Make this section look unique and attractive
3. Latest Resolve Issue Section (show at least 6)
   - Sorted by status of issue
   - For each issue show show card with associate data with view details
   - Clicking on view details it navigate to details page

4. Features Section

    - Show the application features
5. How it work section
6. Extra 2 section
7. Footer
8. 404 Not Found Page
    - Nice error page for wrong URLs
    - Button to go back to Home

All Issues Page (/all-issues)

Show all issues in this page

- Show each issue in the card view with the proper information(image, title, category, status badge, priority badge (High/Normal), Location,Upvote button or icon with total upvote number & View Details button.
- Clicking on view details navigate to issue details page
- Filtering options based on category,status,priority (**before implement check challenge part**)
- Search bar to search the issue (**before implement check challenge part**)
- For upvote button features as follows:

  Logged-in users can **upvote an issue** to show public importance. Each user can upvote an issue **only once**

  Total upvote count is visible on:

    - Issue cards
    - Issue details page

      Rules:

    - If user is not logged in & want to upvote then redirect to login
    - Users cannot upvote their own issue
    - Boosted issues still stay above normal issues
    - After upvote increase upvote count by 1 for that issue in db & show instant update in UI

Issue Details Page (Private Route)

Only logged-in users can visit. Show:

1. Show full information of an issue. Make impressive UI design as you want.
2. Button for:
    - Edit (if logged-in user is submit this issue & status=pending)
    - Delete (if logged-in user is submit this issue)
    - Boost issue priority button if already not boosted (after click it payment to boost the issue per issue 100tk for boost)

      After successful payment the issue priority is set as **high**. Boosted issues appear above normal issues. A tracking record is added to the issue timeline

3. Show staff information if assigned for this issue.

Issue Tracking & Timeline Section

Each Issue Details page must include a **Timeline / Tracking section** that shows the complete lifecycle of the issue.Timeline entries should be **read-only** (cannot be edited or deleted) to preserve audit history.

Timeline Items Should Include:

- Status (Pending / In-Progress / Resolved / Closed)
- Message or note
- Updated by (Admin / Staff / Citizen)
- Date & time

Example Timeline Entries:

- Issue reported by citizen
- Issue assigned to Staff: John Doe
- Work started on the issue
- Issue marked as resolved

- Issue closed by staff

UI Requirements:

- Vertical timeline or stepper UI
- Latest update at the top
- Status badges with colors

Data Flow:

- Every important action must create a timeline entry:
    - Issue creation
    - Staff assignment
    - Status change
    - Boost payment
    - Admin rejection
    - Issue closure

ROLE MANAGEMENT (3 Roles)

1. Admin
    - View all issues
    - Assign staff
    - Reject issues
    - Manage staff
    - Manage citizens
    - View payments
2. Citizen
    - Submit issues
    - Edit/delete own issue(if pending)
    - Boost priority
    - Access premium subscription
    - Track activities
3. Staff

- View assigned issues
- Change issue status
- Add progress updates
- Mark as resolved
- Edit profile

Citizen Dashboard (Private)

Blocked users can log in but **cannot submit, edit, upvote, or boost issues**.
Routes inside dashboard:

1. Dashboard Page:
    - Total issues submitted
    - Total pending issues
    - Total in progress issues
    - Total Resolved issues
    - Total payments
    - Shows these in card stats & charts

2. My Issues Page
    - List all issues from logged-in user
    - Filters option by status, category etc
    - Buttons:
        - Edit issue (if status pending). After clicking it, open a pre-filled modal with issue data where users can edit issues info and submit. After submit must be data update in db & ui instantly
        - Delete issue. After click it user can remove the issue from ui & db
        - View Details. After click it navigate to issue details page

3. Report Issue page:
    - A Form with title, description, category dropdown, upload image, location

- After successfully create an issue save the data in db & navigate to my-issues page
- After create a tracking record is added to the issue timeline
- User Limit:
    1. Free users can report a maximum of 3 issues. If the user not subscribed yet disable to report issue more than 3 and show subscription button which navigate the profile page
    2. If user is premium then user can report unlimited issue
    3. After successfully create issue navigate to my issues page

4. Profile page:

- Logged-in users can see their information and update.
- If the user does not subscribe, the application allows him to subscribe after paying 1000tk. There should be a button for subscribe, after clicking it pay the required amount. After successful payment the user becomes a premium user & there is no limit for issue submit.
- If the user subscribed show premium badge beside their name
- If the user is blocked by the admin show warning & suggest to contact with the authorities

---

Staff Dashboard (Private)

Staff can only view and update issues assigned to them, not all issues.

1. Dashboard Overview
    - Assigned issues count
    - Issues resolved count
    - Today's task
    - Show statistics in charts
    - You can show more stats as well

2. Assigned issues page
    - List of all assigned issues in tabular form. Boosted issues appear above normal issues. Each row has the issue data with an action button

called change status. After clicking the button show dropdown options to change status like in-progress, working, resolved, closed etc. When select any one update the status in db & ui instantly
    - Filtering option based on status, priority etc
    - Staff can change:
        - Pending status into in-progress
        - In-progress status into working
        - Working status into resolved
        - Resolves status into closed
        - While changing the status update the status for that issue in db as well.
        - A tracking record is added to the issue timeline when changing any status
3. Profile page
    - Update image, name
    - Show profile information

Admin Dashboard (Private)

1. Dashboard
    - Total issue count
    - Resolved issue count
    - Pending issue count
    - Rejected issue count
    - Total payment received
    - And so on, you can show these in card stats & charts system as well

      Besides these stats & graphs, show some latest issues, latest some payments, some latest registered users

2. All Issues
    - Show all issues in tabular form. Boosted issues appear above normal issues. Each row will have:
        - Issue title, category, status, priority(normal/high),assigned staff(if already assigned), assigned staff button(if no assigned staff yet)

    - Assign staff button only when not assigned any staff yet. the admin can assign a staff member **only if the issue has no assigned staff**.
      After clicking the **Assign Staff** button, a modal will open with a **dropdown list of all available staff fetched from the database**.
      The admin selects **one staff member** and confirms the assignment.
    - Once a staff member is assigned, the Assign Staff button should be disabled permanently for that issue. Re-assignment is NOT required.

      After assignment:

    -

- The issue's `assigned Staff` is saved in the database & show assigned staff in table
- Issue status remains **pending** (staff has not started yet)
- A tracking record is added to the issue timeline
- The assigned staff will immediately see this issue in their dashboard
- Reject issue button only when status=pending. After clicking it shows confirmation dialogue before rejecting the issue.

  Admin can assign any number of issues for a staff. Issue assignments do not depend on the issue status.

3. Manage users page
    - Show all citizen users in tabular form
    - View subscription info.
    - Block/Unblock button. Admin can block/unblock users from ui & the db as well. Show confirmation dialogue before block/unblock.

4. Manage Staff
    - Add button: When the admin clicks the **Add Staff** button, a modal form opens where the admin must provide the staff's basic information (name, email, phone, photo, password). After submitting the form, the staff account is created both in **Firebase Authentication (email & password)** and in the database with the role set as **staff**. The staff member can then log in to the system using the provided **email and**

password and will be redirected to the **Staff Dashboard**, where they can only view and update issues assigned to them.

- Show all staff's in tabular form
- Each row has update & delete button
- After click update button open modal with pre-filled staff info and admin can update staff information
- Clicking the delete button shows a confirmation button before the delete action happens. After delete remove the staff from the ui & also from db.

⚠️ Important Note

Allowing an admin to create passwords for staff is done **only for assignment simplicity**.
In real-world applications, this approach is **not recommended** due to security risks

5. Payments page
    - Shows all the payments in tabular form
    - Implement filter features
    - Show payments data in chart based on month (**optional, no marks for this**)

6. Profile
    - Admin can view and update their own information.

Login & Registration

- Show nice error messages
- Login: Email/Password + Google Sign-in
- Registration: Name, Email, Password, Photo upload. After successfully registering, store user information in the database.
- No email verification needed

Challenge Tasks (Must Do All)

1. Token Verification & Role-Based Middleware
2. Implement pagination on All-Issues page
3. Search & filter (all-issues page):
    - Implement server-side search by issue name, category,location etc
    - Implement server-side filter by status, priority,category etc
    - Show loader when data fetching from database
4. Add downloadable invoice PDF in admin payments page & user profile page. For reference: How to generate PDF in React using React pdf

---

Optional Tasks (Choose Any 2)

1. Add animations (Framer Motion or AOS)
2. Use Axios interceptors
3. Dark/Light theme
4. Prevent multiple upvotes for logged-in users in the same issue.

Additional Notes

- You can host images anywhere
- Use any CSS (Tailwind, DaisyUI, custom, etc.)
- Deploy client on Firebase Hosting or anywhere

- Deploy server on Vercel or anywhere
- Deploy early!

What to Submit

1. Admin Email:
2. Admin Password:
3. Live Site Link (Client):
4. Client GitHub Repo:
5. Server GitHub Repo:
6. One Staff Email & Password:
7. One citizen Email & Password:

   Try to provide such staff credentials who have already assigned issues & working with issues. Similarly, try to give a user credential who is not a premium user & already posted 2 issues.