

System Design Bugs Fixed - Summary Report

Overview

Successfully fixed all critical bugs in the EM Interview Prep application's system design functionality. The application now has proper categorization, working filters, and clean data.

Issues Fixed

1. System Design Filtering Issue

Problem: Filtering by category, company, and difficulty was not working properly

Root Cause: Mismatch between frontend category names and database category keys

Solution:

- Updated API endpoint to map display names to database keys
- Fixed filtering logic to handle both old and new category formats
- Updated frontend to refetch data when filters change instead of client-side filtering

2. Malformed Questions Cleanup

Problem: 162 questions with nonsensical "Design System N for..." format

Root Cause: Bulk import script generated placeholder questions

Solution:

- Created cleanup script to identify and delete malformed questions
- Removed all 162 malformed questions and their related data
- Preserved only legitimate, well-formed questions

3. Category Redistribution

Problem: 166 questions stuck in old "distributed_systems" category

Root Cause: Old categorization system not updated to new 4-category structure

Solution:

- Implemented intelligent categorization based on question content keywords
- Redistributed all questions into the 4 correct categories:
- Distributed Systems & Infrastructure: 10 questions
- Data & AI/ML Systems: 7 questions
- Real-time & Communication Systems: 7 questions
- Product & Platform Systems: 2 questions

4. API Category Mapping

Problem: Frontend and backend using different category identifiers

Solution:

- Updated system-design-questions API to handle display name mapping
- Updated system-design-categories API to return actual counts from database
- Ensured consistent category handling across all endpoints

5. Database Integrity

Problem: Orphaned records and inconsistent data

Solution:

- Cleaned up all related records (answers, progress, views) for deleted questions
- Ensured referential integrity across all tables
- Updated all question records with proper categories and timestamps

Technical Implementation

Files Modified:

1. `/app/api/system-design-questions/route.ts` - Fixed filtering logic
2. `/app/api/system-design-categories/route.ts` - Dynamic category counts
3. `/components/system-design/system-design-question-bank-client.tsx` - Improved filtering UX
4. `/scripts/fix-system-design-categories.js` - Data cleanup script

Database Changes:

- Deleted 162 malformed questions
- Recategorized 26 remaining questions
- Updated category field and categories array for all system design questions
- Maintained referential integrity across all related tables

Final State

Question Distribution:

- **Total System Design Questions:** 26 (down from 188, removed 162 malformed)
- **Distributed Systems & Infrastructure:** 10 questions
- **Data & AI/ML Systems:** 7 questions
- **Real-time & Communication Systems:** 7 questions
- **Product & Platform Systems:** 2 questions

Functionality Verified:

Category filtering works correctly
Company filtering works correctly
Difficulty filtering works correctly
Search functionality works correctly
Critical questions filter works correctly
No cross-contamination between behavioral and system design questions
All questions are well-formed and meaningful
API endpoints return correct data
Frontend displays questions properly

Testing Results

API Testing:

```
# Category filtering
curl "http://localhost:3000/api/system-design-questions?category=Data%20%26%20AI%20FML%20Systems"
# Returns: 7 questions

curl "http://localhost:3000/api/system-design-questions?category=Real-time%20%26%20Communication%20Systems"
# Returns: 7 questions

curl "http://localhost:3000/api/system-design-questions?category=Distributed%20Systems%20%26%20Infrastructure"
# Returns: 10 questions

curl "http://localhost:3000/api/system-design-questions?category=Product%20%26%20Platform%20Systems"
# Returns: 2 questions

# Company filtering
curl "http://localhost:3000/api/system-design-questions?company=Meta"
# Returns: 2 questions
```

Build Status:

Application builds successfully
No TypeScript errors
No linting errors
All routes functional

Application Status

- **Server:** Running successfully on `http://localhost:3000`
- **Authentication:** Working (Demo login: `mayssam/password123`)
- **System Design Questions:** Fully functional with proper filtering
- **Data Quality:** Clean, well-categorized questions only
- **Performance:** Optimized with proper API filtering

Next Steps

The application is now ready for production use with:

1. Clean, well-categorized system design questions
2. Fully functional filtering system
3. Proper separation between behavioral and system design content
4. Robust API endpoints with proper error handling
5. Responsive UI with good user experience

All critical bugs have been resolved and the system design functionality is working as intended.