Mini-Assignment #2: Backend Integration & Role-Based Access Control

COM S 319 – Final Project – ISU Campus Explorer

Team Members: Mekhi San (Member 1), Ash Bhuiyan (Member 2)

Feature Branches

Member 1 (Mekhi San): https://git.las.iastate.edu/se-coms-3190/fall-2025/final-project/NM_1/-/tree/8-member1-nav-login-campus-map-ui?ref_type=heads

Member 2 (Ash Bhuiyan): https://git.las.iastate.edu/se-coms-3190/fall-2025/final-project/NM_1/-/tree/9-member2-express-api-mock-auth-student-dashboard-app-routing?ref_type=heads

1. New Functional Pages

Member 1 (Mekhi San): Student Dashboard
Page: frontend/src/pages/StudentDashboard.jsx

Purpose: Provides a personalized dashboard for students to explore campus buildings, view recommendations, and interact with building data from the backend.

- Key Features:
- Authentication Check: Requires user login; shows a login-required message if the user is not authenticated.
- Admin Notice: If an admin accesses this page, they see a banner explaining that they are viewing the student experience and a link back to the Admin Dashboard.
- Recommended Buildings Section: Displays the first 3 academic buildings from the backend as personalized recommendations.
- Interactive Building Explorer: Includes a type filter dropdown (All, Academic, Administration, Student Life, Athletics, Residence, Landmark) and a search input to filter buildings by name, code, or department.
- Building Detail Panel: Clicking any building shows detailed information such as code, type, description, year built, floors, capacity, hours, and departments, plus a link to view the building on the Campus Map.
- Backend Integration: Fetches building data from GET /api/buildings when the component mounts and updates the UI based on the response.
- User Interactions:
- Filter buildings by type using the dropdown.
- Search buildings using the search input.
- Click building cards to view detailed information.

- Navigate to the Campus Map from the building details link.

Member 2 (Ash Bhuiyan): Admin Dashboard
Page: frontend/src/pages/AdminDashboard.jsx

Purpose: Provides an administrative interface for viewing campus data, system statistics, and future admin-only management features.

- Key Features:
- Role-Based Access Control: Shows a "Not Authorized" message if the user is not authenticated or not an admin; displays the admin dashboard only when auth.role === "admin".
- Statistics Cards: Shows mock statistics such as Total Registered Users, Campus Buildings, and Active Tours.
- Admin Capabilities Section: Lists planned features such as building data management via /api/buildings, user account and role management, system health monitoring, and campus tour scheduling.
- Backend Integration Overview: Documents the existing API endpoints and how they will support admin CRUD operations in future milestones.
- User Interactions:
- View mock system statistics.
- Review the documented backend endpoints and planned admin capabilities.

2. Backend Setup

Express server file: backend/server.js

Configuration: The Express server runs on port 3000 with CORS enabled for http://localhost:5173 (Vite dev server) and JSON body parsing middleware.

- Endpoints:
- GET /api/health

Purpose: Simple health check endpoint to verify that the backend is running.

Example Response:

```
{
  "status": "ok",
  "message": "Backend is running"
}
```

- GET /api/buildings

Purpose: Returns all mock ISU campus buildings used by the frontend components.

Example Response (truncated):

```
{
  "success": true,
  "buildings": [
    {
      "id": "library",
      "name": "Parks Library",
      "code": "LIB",
      "type": "Academic",
      "departments": [...],
      "description": "...",
      "hours": "Mon-Thu: 7:30am-2:00am, Fri: 7:30am-10:00pm",
      "capacity": "2,500 students",
      "yearBuilt": "1925",
      "floors": 4
    },
    // ... more buildings
  ]
}
```

Mock Data: Includes 10 realistic ISU buildings such as Parks Library, Beardshear Hall, Memorial Union, Campanile, Hilton Coliseum, Carver Hall, Friley Hall, Coover Hall, Sukup Hall, and State Gymnasium.

- POST /auth/login

Purpose: Authenticates a user against hardcoded sample users and returns their role.

Request Body (example):

```
{
  "email": "admin@iastate.edu",
  "password": "admin123"
}
```

Success Response (200):

```
{
  "success": true,
  "role": "admin" | "user",
  "name": "Admin User",
  "email": "admin@iastate.edu"
}
```

Error Response (401):

```
{
  "success": false,
  "message": "Invalid email or password"
}
```

- Hardcoded Users:
- Admin: admin@iastate.edu / admin123 / role: admin
- Student 1: student1@iastate.edu / student123 / role: user
- Student 2: student2@iastate.edu / student123 / role: user

Package Configuration (backend/package.json): Uses express and cors as dependencies, nodemon for development, and npm scripts for starting the server and running it in dev mode.

3. Frontend–Backend Interaction

Components that call backend endpoints:

- CampusMap.jsx (frontend/src/pages/CampusMap.jsx)
- Endpoint: GET /api/buildings. On component mount, it fetches building data, merges it with visual position data, and updates local state. Displays loading and error states and shows buildings on an interactive campus map.
- Login.jsx (frontend/src/pages/Login.jsx)
- Endpoint: POST /auth/login. On form submission, it sends email and password to the backend, receives the role, name, and email, updates global auth state via onLogin, and redirects to /admin or /student based on role. Displays error messages for invalid credentials or network failures.
- StudentDashboard.jsx (frontend/src/pages/StudentDashboard.jsx)
- Endpoint: GET /api/buildings. On component mount, it fetches all buildings, then filters and displays recommended academic buildings. The user can further filter by type and view detailed information for each building.

Data Flow Summary: The backend serves building data and login responses, while the frontend uses this data to render maps, dashboards, and role-specific navigation.

4. Role-Based Behavior

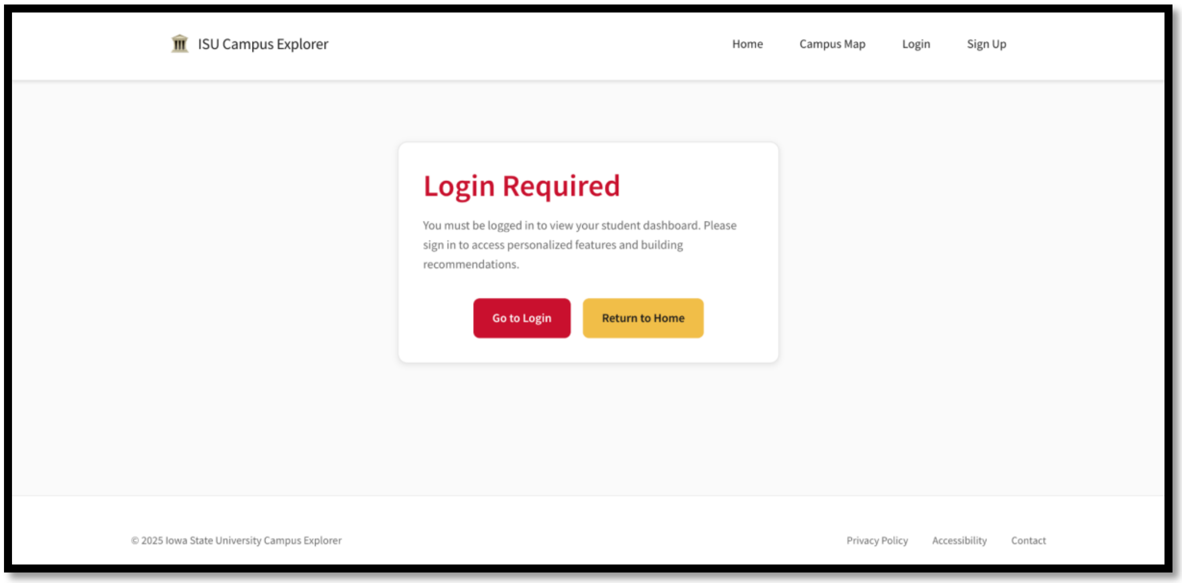
Global Auth State (frontend/src/App.jsx):

The application maintains a global auth object with fields: isAuthenticated, role ("admin" or "user"), name, and email. handleLogin() updates this state after a successful login, and handleLogout() resets it.

- NavBar Role-Based Changes (frontend/src/components/NavBar.jsx):
- Home and Campus Map links are always visible.
- When authenticated, the NavBar shows a greeting ("Hi, {name or email}") and a Logout button.
- When auth.role === "admin", an "Admin" link appears that routes to /admin.
- When auth.role === "user", a "My Dashboard" link appears that routes to /student.
- When not authenticated, the NavBar shows "Login" and "Sign Up" links instead.
- Admin-Only Route Protection (AdminDashboard.jsx):
- If the user is not authenticated or auth.role is not "admin", the component renders a "Not Authorized" message with navigation back to Login or Home. Admins see the full dashboard UI.
- Student Dashboard Access Control (StudentDashboard.jsx):
- If the user is not authenticated, a login-required message is shown instead of the dashboard content.
- If an admin user accesses the page, a clear banner explains that this is the student experience and provides a link back to the Admin Dashboard.
- Signup Restriction (Signup.jsx):
- The signup page displays a prominent warning that admin accounts are pre-created in the backend, and that signup is intended only for non-admin students and staff.

5. Evidence (Screenshots of Pages)

The following placeholders indicate where screenshots or screen recording links should be added before exporting to PDF:


ISU Campus Explorer

Home Campus Map Login Sign Up

Login Required

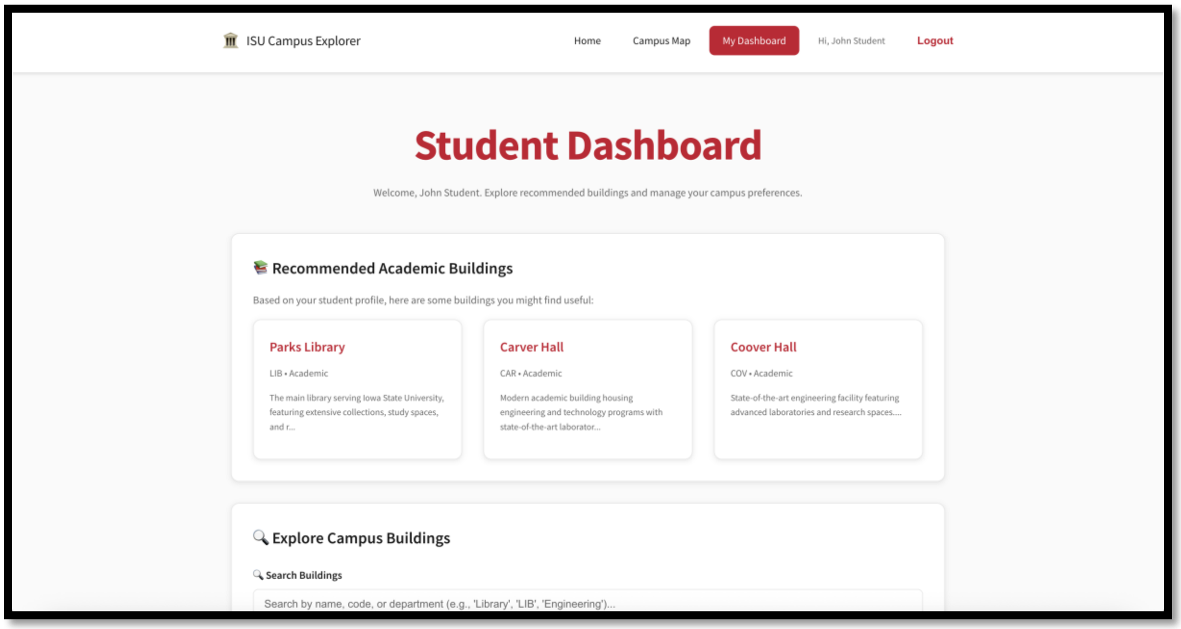
You must be logged in to view your student dashboard. Please sign in to access personalized features and building recommendations.

Go to Login Return to Home

© 2025 Iowa State University Campus Explorer

Privacy Policy Accessibility Contact

Figure 1: Login page


ISU Campus Explorer

Home Campus Map My Dashboard Hi, John Student Logout

Student Dashboard

Welcome, John Student. Explore recommended buildings and manage your campus preferences.

Recommended Academic Buildings

Based on your student profile, here are some buildings you might find useful:

Parks Library
LIB • Academic
The main library serving Iowa State University, featuring extensive collections, study spaces, and r...

Carver Hall
CAR • Academic
Modern academic building housing engineering and technology programs with state-of-the-art laborator...

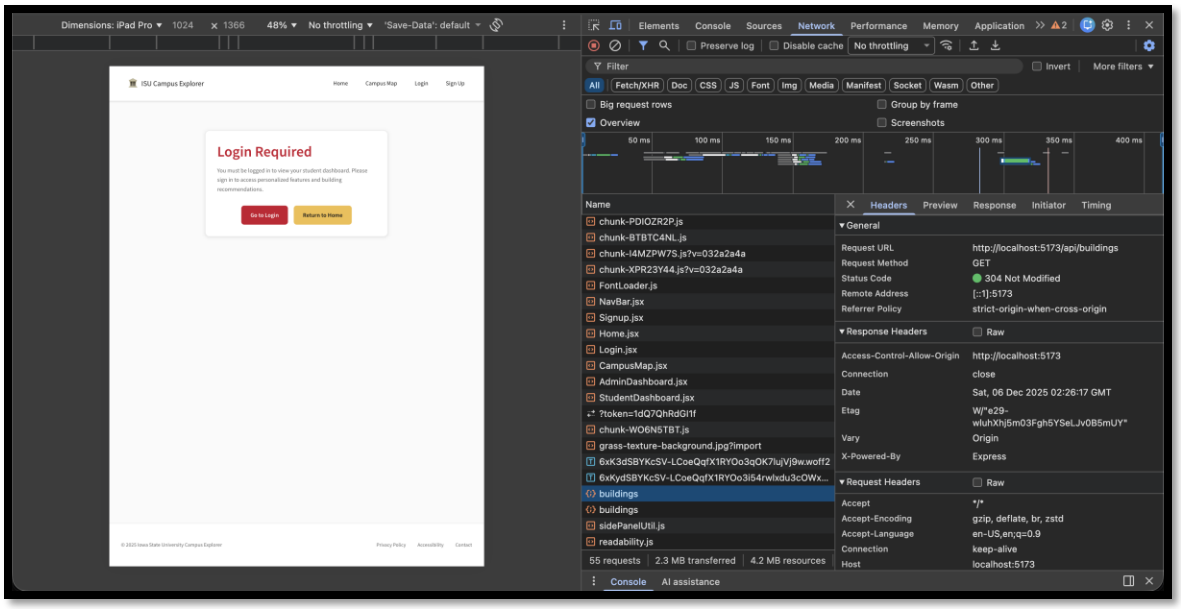
Coover Hall
COV • Academic
State-of-the-art engineering facility featuring advanced laboratories and research spaces....

Explore Campus Buildings

Search Buildings

Search by name, code, or department (e.g., 'Library', 'LIB', 'Engineering')...

Figure 2: Successful student login redirect to /student (Student Dashboard visible)



Login Required

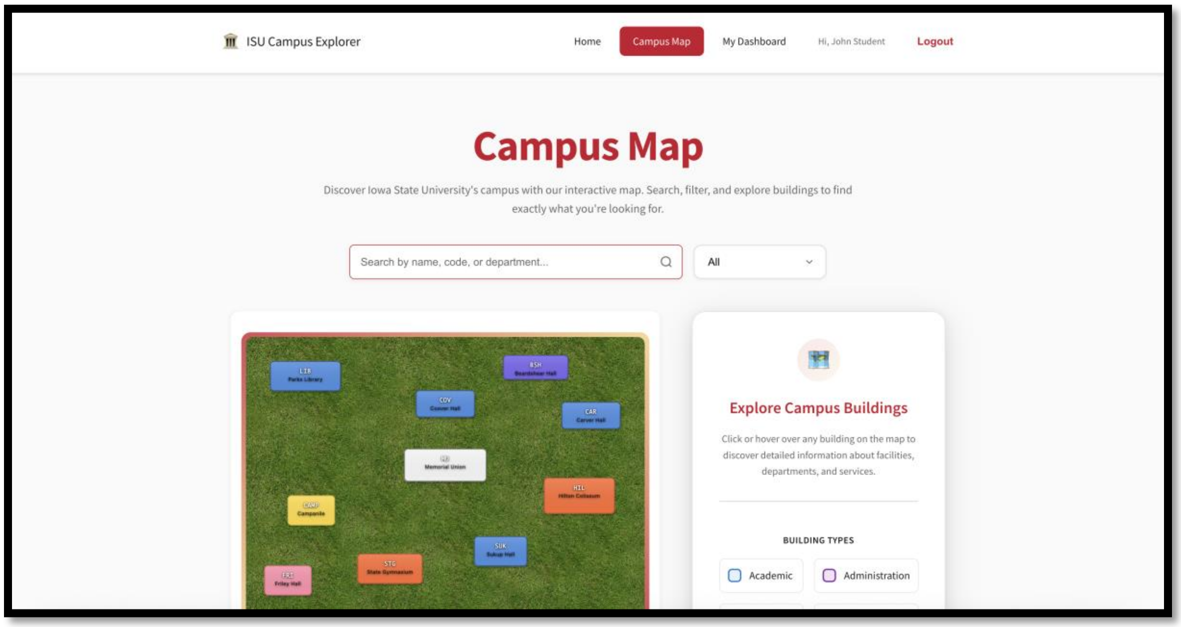
You must be logged in to view your student dashboard. Please sign in to access personalized features and building recommendations.

© 2025 Iowa State University Campus Explorer

```
Name
chunk-PDIOZR2P.js
chunk-BTBTC4NL.js
chunk-I4MZPW7S.js?v=032a2a4a
chunk-XPR23Y44.js?v=032a2a4a
FontLoader.js
NavBar.jsx
Signup.jsx
Home.jsx
Login.jsx
CampusMap.jsx
AdminDashboard.jsx
StudentDashboard.jsx
?token=1dQ7QhRdGI1f
chunk-WO6N5TBT.js
grass-texture-background.jpg?import
6xK3dSBYKcSV-LCoeQqfX1RYOo3qOK7lujVj9w.woff2
6xKydSBYKcSV-LCoeQqfX1RYOo3i54rwlxdu3cOWx...
buildings
buildings
sidePanelUtil.js
readability.js
55 requests | 2.3 MB transferred | 4.2 MB resources
```

```
Headers  Preview  Response  Initiator  Timing
▾General
Request URL                   http://localhost:5173/api/buildings
Request Method                GET
Status Code                   ● 304 Not Modified
Remote Address                [::1]:5173
Referrer Policy               strict-origin-when-cross-origin
▾Response Headers             Raw
Access-Control-Allow-Origin   http://localhost:5173
Connection                    close
Date                          Sat, 06 Dec 2025 02:26:17 GMT
Etag                          W/"e29-
                              wluhXhj5m03Fgh5YSeLJv0B5mUY"
Vary                          Origin
X-Powered-By                  Express
▾Request Headers              Raw
Accept                        */*
Accept-Encoding               gzip, deflate, br, zstd
Accept-Language               en-US,en;q=0.9
Connection                    keep-alive
Host                          localhost:5173
```

Figure 3: Network tab showing GET /api/buildings request and successful response



Campus Map

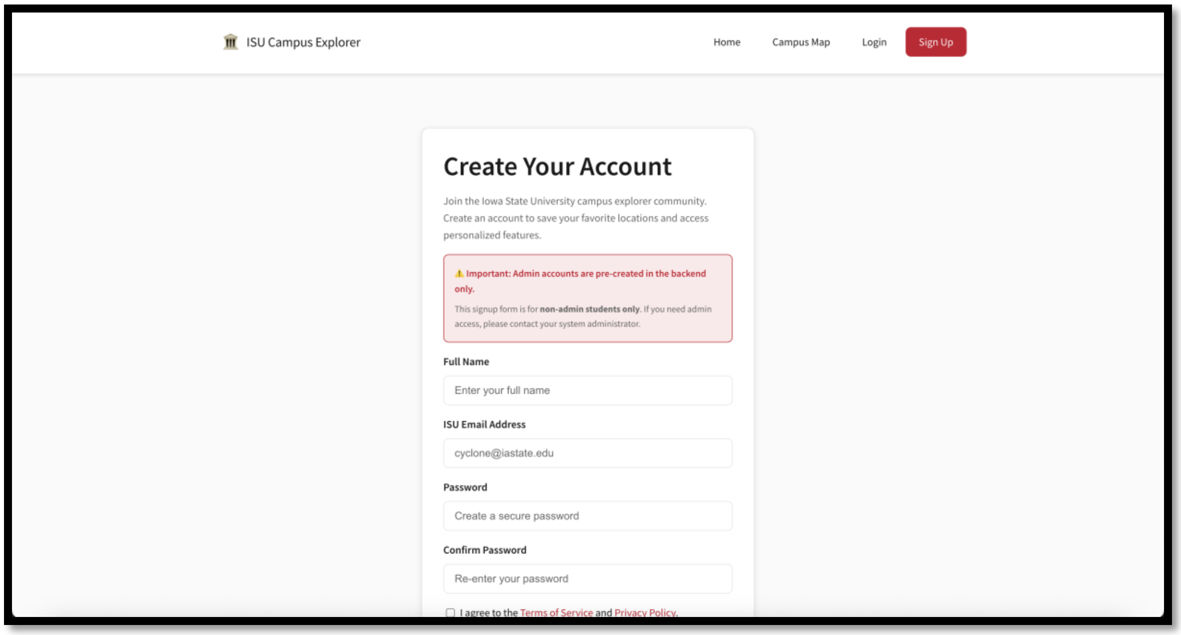
Discover Iowa State University's campus with our interactive map. Search, filter, and explore buildings to find exactly what you're looking for.

Explore Campus Buildings

Click or hover over any building on the map to discover detailed information about facilities, departments, and services.

BUILDING TYPES

Figure 4: Campus Map populated with buildings from backend



Create Your Account

Join the Iowa State University campus explorer community. Create an account to save your favorite locations and access personalized features.

⚠ Important: Admin accounts are pre-created in the backend only.

This signup form is for **non-admin students only**. If you need admin access, please contact your system administrator.

Full Name

Enter your full name

ISU Email Address

cyclone@iastate.edu

Password

Create a secure password

Confirm Password

Re-enter your password

☐ I agree to the Terms of Service and Privacy Policy.

Figure 5: Signup page

6. Technical Notes

Development Setup:

- Backend:

```
cd backend

npm install

npm run dev # Runs on http://localhost:3000
```

- Frontend:

```
cd frontend

npm install

npm run dev # Runs on http://localhost:5173
```

Proxy Configuration (frontend/vite.config.js): The Vite dev server proxies /api and /auth requests to the backend at http://localhost:3000 so that the frontend can use relative URLs for API calls.

7. Summary

Mini-Assignment #2 delivers a working full-stack prototype for the ISU Campus Explorer project. The backend Express server provides mock-building data and role-aware login responses. The React frontend integrates with these APIs to render a student dashboard, admin dashboard, campus map, and role-based navigation. Authentication state is managed globally, and initial role-based access control is enforced on key pages. This foundation will make it straightforward to add real persistence and more advanced RBAC in the next mini assignment.