Mini-Assignment #1: Initial Planning & UI Setup

Project: ISU Campus Explorer
Team: NM_1
Team Members: Ash Bhuiyan & Mekhi San

1. Project & Page Overview

ISU Campus Explorer is a single-page application designed to help new students and visitors navigate Iowa State University. In the final project, the app will combine a map/list explorer, rich building details, tour planning, and role-based access control. For Mini-Assignment #1, we focused only on the frontend shell: setting up React, React Router, and the initial static pages that will later connect to our Node/Express backend.

2. Frontend Setup Process

- Initialized the frontend React app inside the frontend/ directory and verified the standard Vite/React structure.
- Installed *react-router-dom* and wrapped the root of the app in *<BrowserRouter>* inside *src/main.jsx*.
- Created *src/App.jsx* as the main application shell, including the Routes configuration and a shared footer.
- Built a shared *NavBar* component with navigation links to *Home*, *Campus Map*, *Login*, and *Signup* pages for smooth transitions.
- Added a global styles.css file to define our design system (colors, typography, layout helpers, buttons, cards, and form styling).
- Verified navigation between all pages and ensured everything runs as a static UI with no backend dependencies yet.

3. File Ownership Summary (Mini-Assignment #1)

To simulate real-world collaboration, we divided the initial UI work and files evenly. Each teammate was responsible for designing, implementing, and testing their pages, while shared infrastructure (routing and layout) was coordinated together.

Ash Bhuiyan – Primary Responsibilities:

- *src/pages/Home.jsx* – Landing page that introduces ISU Campus Explorer and highlights key features (map explorer, building directory, accessibility, and mobile support).
- *src/pages/CampusMap.jsx* – Static prototype of the campus map explorer with a simplified grid-based campus, building cards, and filters by building type and search term.
- *src/styles.css* – Core visual design system: layout grid, typography scales, buttons, cards, form styling, and responsive behavior.
- High-level layout planning for how Home and Campus Map will connect to later features like Building Detail and Tour Planner.

Mekhi San – Primary Responsibilities:

- *src/pages/Login.jsx* – Static login form with local state, client-side validation for email and password, and clear error messages to guide the user.
- *src/pages/Signup.jsx* – Static signup form for new non-admin users, including name, ISU email (@iastate.edu), password confirmation, and a required terms-of-service checkbox.
- *src/components/NavBar.jsx* – Top navigation bar with brand title and links to Home, Campus Map, Login, and Signup; uses React Router links to highlight the active page.
- *src/main.jsx* – Application entry point that renders *<App />* inside *<BrowserRouter>*, connecting the routing system to the DOM root.

4. Page Purpose & Intended Interaction

Home Page (Ash): The *Home.jsx* page acts as the main landing screen for the SPA. It uses a hero section with a badge, large heading, descriptive lead text, and two primary call-to-action

Buttons: One to open the *Campus Map* and another to go to *Sign-up*. Below the hero, feature cards explain what the app offers (campus map, building directory, accessibility, mobile support).

This page gives new users a clear starting point and communicates the project's goals visually.

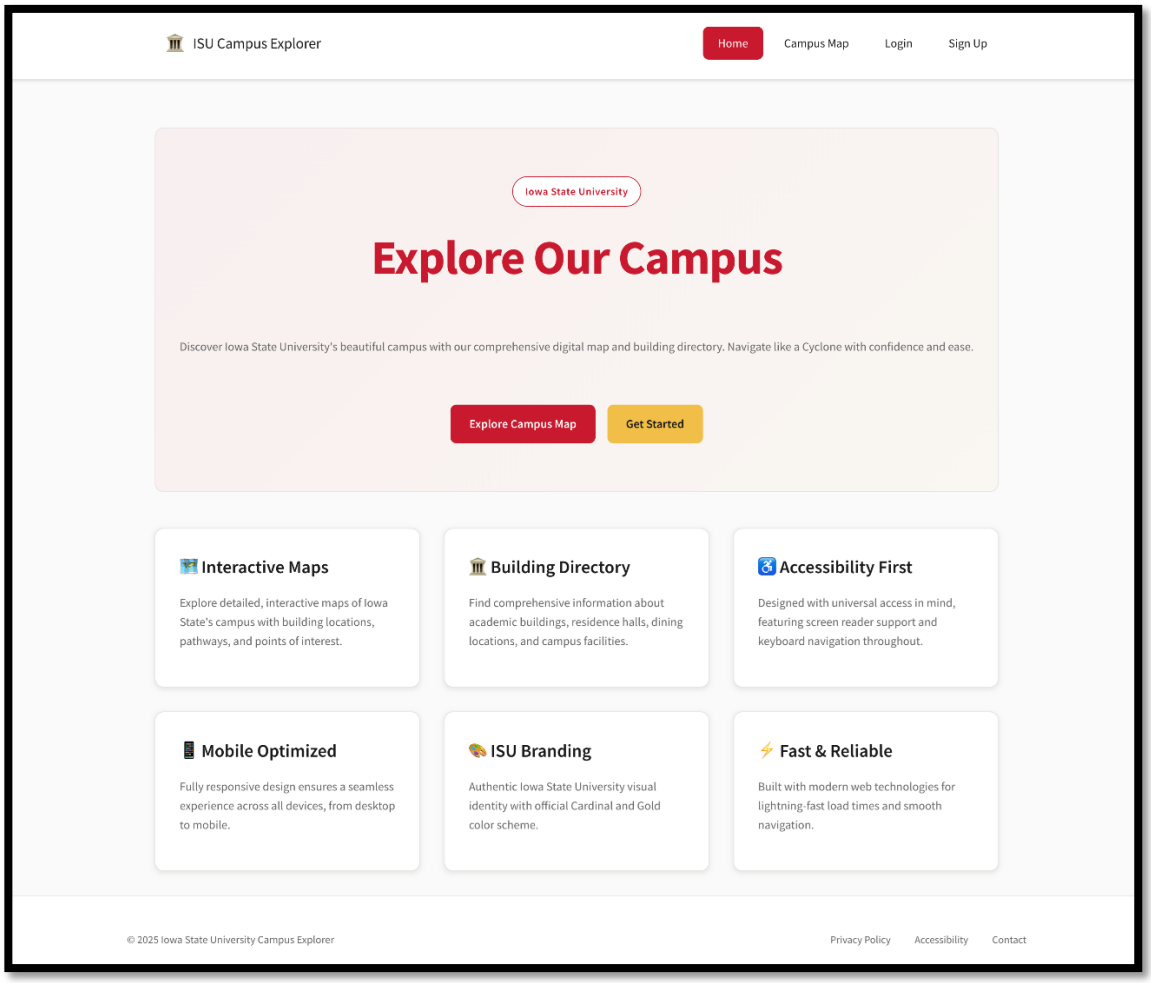
Campus Map Page (Ash): *CampusMap.jsx* is a static prototype of the future map/list explorer. For now, it renders a simplified map grid with building shapes, hover and selection states, and a side panel showing building details (description, hours, departments, capacity, and year built). A search input and building-type filter dropdown simulates how users will narrow results. This page lays out the structure for the final interactive map.

Login Page (Mekhi): *Login.jsx* provides a structured form for returning users. The form tracks email and password using React state and runs client-side validation (valid email format and minimum password length). Any validation errors are surfaced inline under each field with ARIA attributes for accessibility. The form is wrapped in a centered card layout to match our visual system.

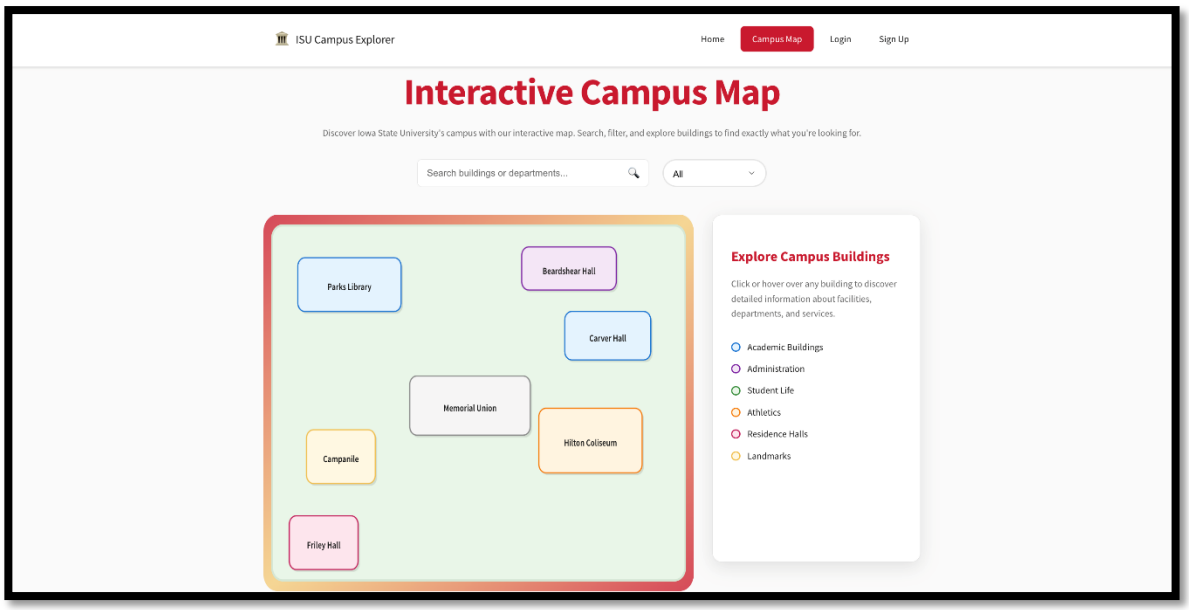
Signup Page (Mekhi): *Signup.jsx* mirrors the login layout but adds additional fields (full name, ISU email, password, password confirmation, and terms checkbox). It enforces an '*@iastate.edu'* email pattern and matching passwords, displaying errors inline. Once hooked to a backend, this page will create non-admin user accounts for the system.

5. Screenshots of Pages

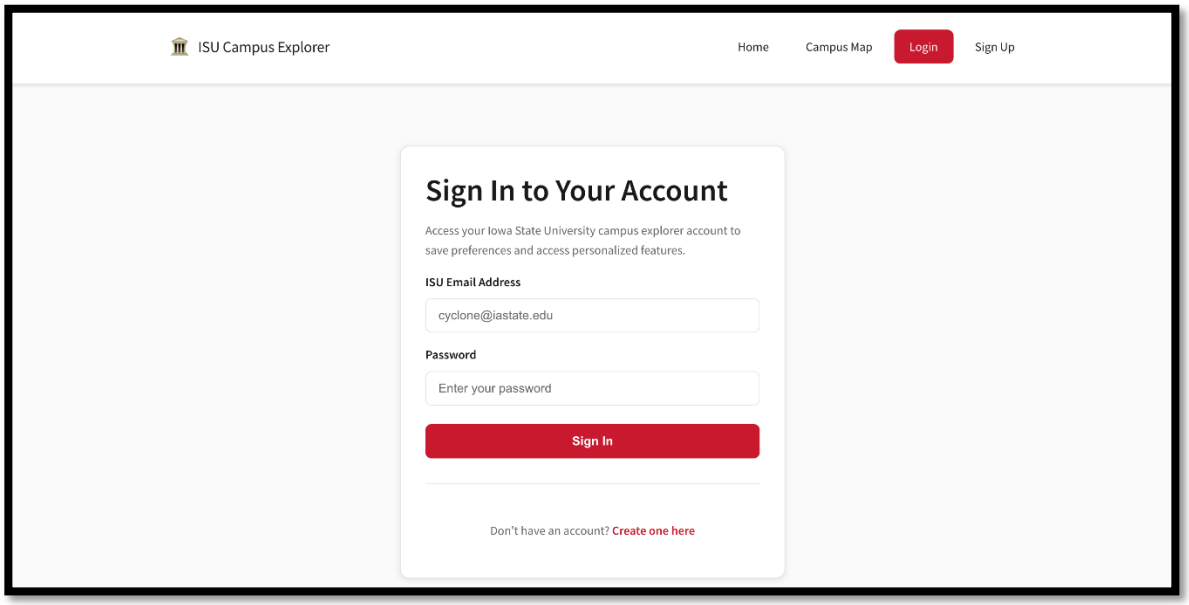
- **Home:**



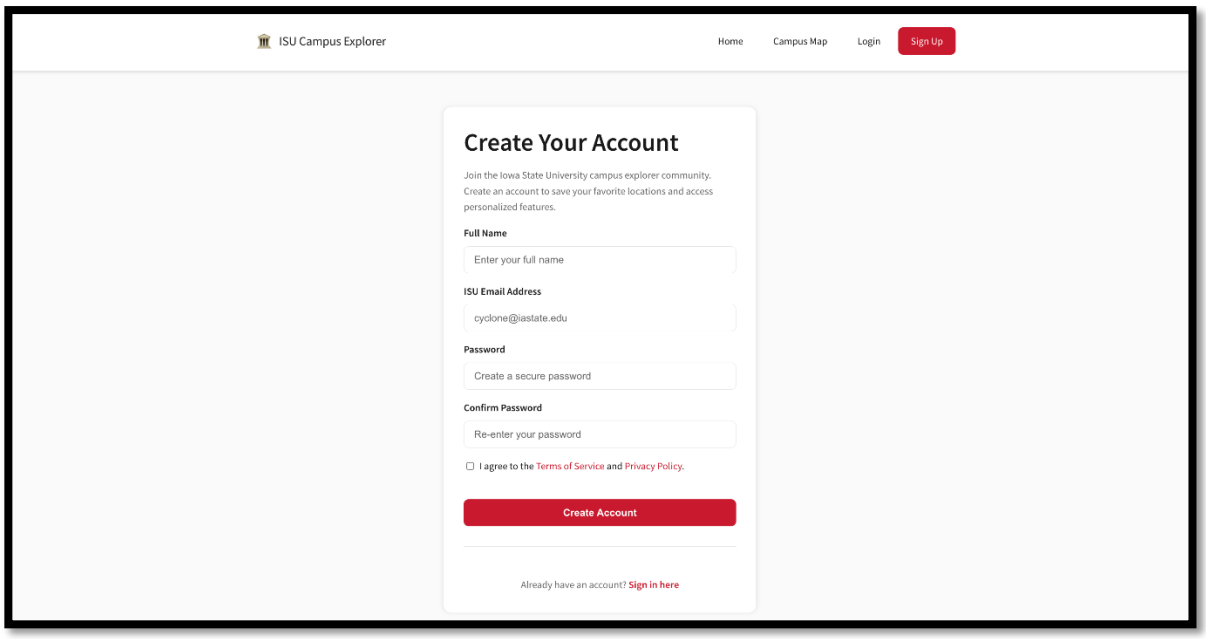
- **Campus Map:**



- **Login:**



- **Sign-up:**



6. Planning Notes & Next Steps

Planning: For this Mini-Assignment, we sketched the layout of each page based on our Final Project Proposal wireframes (Home/Map Explorer, Login/Signup, and Confirmation). We identified which information needs to appear above the fold on the Home page, how navigation should behave, and what fields are required for authentication. The Campus Map layout was designed to mirror the final map/list hybrid: a visual map area alongside a detail panel and filters. Next

Steps: In later assignments we will connect these static pages to a Node/Express backend and a real database. Login and Signup will use JWT-based sessions with role-based access control, while Campus Map will fetch buildings and tours from the API. The structure we created in Mini-Assignment #1 ensures that these integrations can be added without rewriting the UI.