

Cyclone Compass

Final Project Technical Report

SE/COM S 3190 – Construction of User Interfaces
Spring 2025

Team Members:

Maxim Duncan
mgdunc@iastate.edu

Shivadhar Reddy Maddi
msreddy@iastate.edu

May 11, 2025

1. Introduction

Cyclone Compass is a full-stack web application created to enhance the onboarding experience for new and prospective Iowa State University students. Inspired by our challenges navigating campus, this original project serves as a virtual guide to student life. It targets those who may not have the opportunity to physically tour the campus, offering them insights into dorms, clubs, dining, and more.

2. Project Description

Cyclone Compass is designed with a modular layout, featuring:

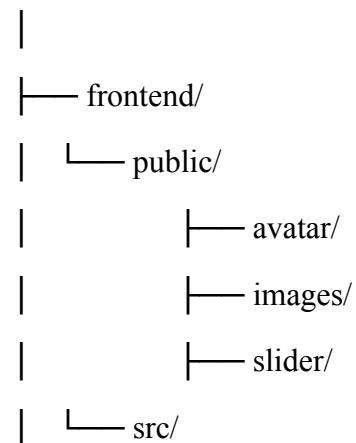
- **Home Page** – Overview and navigation.
- **Campus Map Tour** – Interactive hotspots with descriptions.
- **Dorm Explorer** – Information and images of residence halls.
- **Dining Guide** – Locations, menus, and student reviews.
- **Clubs & Events** – Filterable list of student activities.
- **T-Shirt Store** – Users can browse and purchase ISU-themed merchandise.

Each section uses CRUD operations:

- Admins manage content (Create, Read, Update, Delete).
- Regular users can view and in some cases submit content (e.g., reviews).

3. File and Folder Architecture

cyclonecompass/



```
|   └── api/
|   └── assets/
|   └── components/
|   └── context/
|   └── pages/
|   └── App.jsx
|   └── main.jsx
|
└── backend/
    ├── api/
    ├── config/
    └── server.js
|
└── documents/
    ├── Mini-Assessments/
    └── PS1_final_proposal.pdf
```

4. Code Explanation and Logic Flow

4.1. Frontend–Backend Communication

The frontend uses Axios to make RESTful API requests to the Express backend. Routes are organized under /api endpoints, and JSON responses are handled using async/await patterns.

4.2. React Component Structure

React pages are structured under src/pages/, and reusable components live under src/components/. Props and local state are passed down for rendering dynamic data. useEffect hooks trigger data fetches.

4.3. Database Interaction

The backend connects to MongoDB using **Mongo Client**. Models define schemas for dorms, dining clubs, reviews, and t-shirt products. API routes perform CRUD operations, and controllers validate data.

4.4. Code Snippets

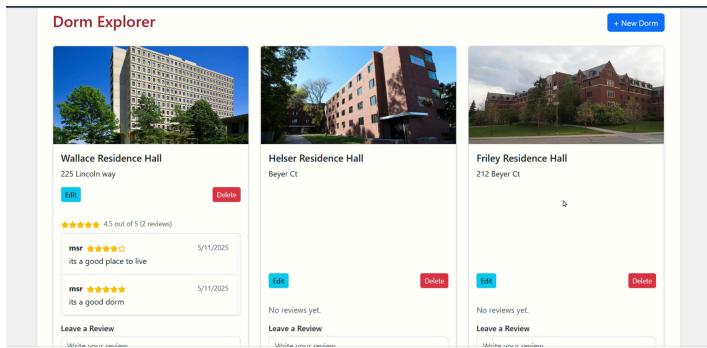
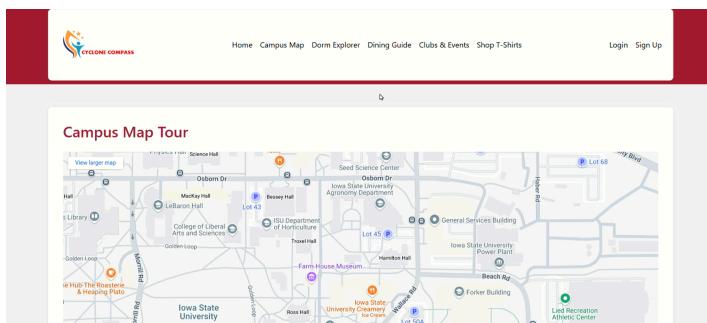
1. Backend Route (dorms.js)

```
// GET all dorms
router.get('/', async (req, res) => {
  try {
    const dorms = await Dorm.findAll();
    res.json(dorms);
  } catch (err) {
    console.error('GET /api/dorms error:', err.message);
    res.status(500).json({ error: err.message });
  }
});
```

2. Mongoose Model (Dorm.js)

```
// Create a new dorm
static async create(data) {
  const now = new Date();
  const doc = {
    name: data.name,
    description: data.description || '',
    img: data.img || '',
    createdAt: now,
    updatedAt: now
  };
  const result = await this._coll().insertOne(doc);
  return { _id: result.insertedId, ...doc };
}
```

5. Web View Screenshots and Annotations



6. Installation and Setup Instructions

1. Clone Repository

```
git clone https://gitlab.com/your-repo-name
```

```
cd cyclonecompass
```

2. Backend Setup

```
cd backend
```

```
npm install
```

```
# Create ` `.env` file with:
```

```
# MONGO_URI=<your_mongo_uri>
```

```
# PORT=5000
```

```
npm run dev
```

3. Frontend Setup

```
cd frontend
```

```
npm install
```

```
npm run dev
```

Visit: <http://localhost:5173>

7. Contribution Overview

Login & Signup Max Duncan

Campus Map Tour Max Duncan

Dorm Explorer Max Duncan

Dining Guide Shivadhar Maddi

Clubs & Events Shivadhar Maddi

Buy a T-Shirt Shivadhar Maddi

Admin login Shivadhar Maddi

8. Challenges Faced

1. Had trouble connecting frontend with backend

- Solved by redoing backend code

2. Could not figure out admin tasks

- Asked TA for inspiration

9. Final Reflections

Building Cyclone Compass taught us full-stack development in a realistic team environment. We strengthened our skills in React, Express, and MongoDB, and learned how to debug and collaborate effectively. In the future, we'd implement more advanced user roles, improve styling, and deploy the app for real-world use.