

Goal

We want to create an employee/HR website portal for managing the new employee onboarding process. Employees will be able to update personal information, upload required documents for identification and work authorization, and monitor their housing situation.

HR will be able to access employee information and make changes regarding the onboarding status or housing assignments.

Requirements

1. Coding
 - a. Backend: Node.js Express server
 - i. The server must be organized based on the MVC architecture.
 - b. Frontend:
 - i. Employee Side: React, Redux, [Material UI](#)
 - ii. HR Side: Angular, NgRx, [Angular Material UI](#)
 - c. Database: MongoDB
 - d. Cloud: AWS S3
 - i. All file uploads should be stored on S3. The *server* should be interacting with S3.
2. Leads: Set up your team's GitHub repository and add the Training Manager as collaborators.
 - a. josie.yan@beaconfireinc.com
 - b. cate.zeng@beaconfireinc.com
3. Teams
 - a. Leads will be in charge of deciding on the project design (UI, organizing the project file structure), assigning workloads, and making sure everything progresses smoothly.
 - b. The Git repository should follow good practices regarding branches, commits, merging, pull requests, etc.
 - c. Leads will participate in daily stand-up meetings with the Training Managers. They will report about the following:
 - i. What features have you implemented so far?
 - ii. What issues/bugs have you encountered?
 - iii. What is each member in charge of, and what is their progress?
 - iv. (optional, if anything is new) project demo

Features

<u>Employees(React)</u>	<u>HR(Angular)</u>
Login Navigation bar	
Registration	Employee Profiles
Onboarding Application	Visa Status Management
Personal information	Hiring Management
Visa Status	Housing Management
Housing	

For Employees (**React**)

1. Registration page
 - a. An HR representative must generate a registration token and a link to the registration page, then send it to a new employee's email. This is the only way for new employees to access the registration page. See **HR #Hiring Management**.
 - i. It should be able to send an actual email message to a valid email address. Here are some options for sending emails: EmailJS (recommended), nodemailer
 - b. Users must provide a unique username, password, and unique email
 - i. On successful registration, users should be automatically and randomly assigned to a house. See **HR #Housing Management**.
2. Login page
 - a. Users should provide their username and password to log in, after which they are redirected to the Personal Information page.
 - b. If they are not logged in, they **only** have access to the login page.
 - c. If logged in, users can access the entire website.
 - d. If the user has an active session or valid token, then after refreshing the page, they should still be logged in. (Hint: use localStorage/sessionStorage/cookie)
3. Onboarding Application page
 - a. Onboarding applications can have one of three statuses: Not Started, Pending, Approved, Rejected.
 - b. Users should be redirected to this page after they log in, only in the following cases:
 - i. **Not Started**: they'll fill in the application fields and submit for the first time
 - ii. **Rejected**: they'll be able to view feedback on why their application was rejected, make changes, and resubmit. Once he/she resubmits, the status would be pending again.

- iii. **Pending:** they've submitted the application, so the page should say "Please wait for HR to review your application." They can view the submitted application (not editable) and a list of all documents that they've uploaded
 - 1. They should be able to download each document.
 - 2. They should be able to open a preview of the document in the browser.
 - iv. If approved, they should be redirected to the home page.
 - c. Input fields (bold fields are **required**, and the rest are optional for the user)
 - i. **First name, last name**, middle name, preferred name
 - ii. Profile picture (with a default placeholder)
 - iii. **Current address** (building/apt #, street name, city, state, zip)
 - iv. **Cell phone number**, work phone number
 - v. Car information (make, model, color)
 - vi. Email (pre-filled based on email that received registration token, cannot be edited)
 - vii. **SSN, date of birth**, gender (male, female, i do not wish to answer)
 - viii. **"Are you a citizen or permanent resident of the U.S?"**
 - 1. Yes: choose **"Green Card"** or **"Citizen"**
 - 2. No: **"What is your work authorization?"** (require them to upload files, you can test it with blank pdfs)
 - a. H1-B, L2, F1(CPT/OPT), H4, Other
 - b. If F1(CPT/OPT): show an input field for uploading their OPT Receipt. See **Employee #Visa Status Management**.
 - c. If other: show an input box to specify the visa title
 - d. Start and end date
 - ix. **"Do you have a driver's license?"**
 - 1. Yes: **driver's license number, expiration date**, and an **uploaded copy of the license**
 - x. Reference (who referred you to this company? There can only be 1)
 - 1. **First name, last name**, middle name, **phone, email, relationship**
 - xi. **Emergency contact(s) (1+)**
 - 1. **First name, last name**, middle name, **phone, email, relationship**
 - d. Summary of uploaded files or documents (if applicable)
 - i. Profile picture
 - ii. Driver's license
 - iii. Work authorization
4. Navigation bar
 - a. **Personal Information, Visa Status Management, Housing**
5. Personal Information page (organized into sections)
 - a. The following sections should have a button that enables users to edit the fields. Clicking the edit button would replace it with a "Cancel" and "Save" button. If they

click “Cancel”, the website should ask if they want to discard all of their changes, and if they click “Yes”, all changes should be undone.

- b. Name
 - i. First name, last name, middle name, preferred name
 - ii. Profile picture
 - iii. Email
 - iv. SSN, date of birth, gender
 - c. Address
 - i. Building/apt #, street name, city, state, zip
 - d. Contact Info
 - i. Cell phone number, work phone number
 - e. Employment
 - i. Visa title, start date, end date
 - f. Emergency contact
 - i. First name, last name, middle name, phone, email, relationship
 - g. Documents
 - i. The list of uploaded documents (driver’s license, work authorization).
 - ii. They should be able to download each document.
 - iii. They should be able to open a preview of the document in the browser.
6. Visa Status Management page
- a. Users should be able to manage their work authorization documents. For this project, we’re focusing on OPT visas, so the following only applies if the user selected OPT for their onboarding application. Otherwise, the page should not show any of these documents. This page is used to track the status of their uploaded documents and required next steps.
 - b. Documents should be uploaded in this order, one by one. Users can only upload the next document after the previous document has been approved by HR.
 - i. OPT Receipt (submitted in onboarding application, waiting for approval)
 - 1. If pending, there should be a message “Waiting for HR to approve your OPT Receipt”.
 - 2. If approved, there should be a message “Please upload a copy of your OPT EAD”.
 - 3. If rejected, users should see HR’s feedback. See **HR #Visa Status Management**.
 - ii. OPT EAD
 - 1. If pending, there should be a message “Waiting for HR to approve your OPT EAD”.
 - 2. If approved, there should be a message “Please download and fill out the I-983 form”.
 - 3. If rejected, users should see HR’s feedback. See **HR #Visa Status Management**.
 - iii. I-983

1. There should be two documents available for download, "Empty Template" and "Sample Template" (can be blank pdfs). There must also be a button to upload the filled out form.
 2. If pending, there should be a message "Waiting for HR to approve and sign your I-983".
 3. If approved, there should be a message "Please send the I-983 along with all necessary documents to your school and upload the new I-20".
 4. If rejected, users should see HR's feedback. See **HR #Visa Status Management**.
- iv. I-20
1. If pending, there should be a message "Waiting for HR to approve your I-20".
 2. If approved, there should be a message "All documents have been approved".
 3. If rejected, users should see HR's feedback. See **HR #Visa Status Management**

7. Housing

- a. **House Details:** Users should be able to view their assigned housing details. They are unable to modify anything.
 - i. Address
 - ii. List of roommates (preferred or legal full name and phone number)
- b. **Facility Reports:** Users should be able to report facility issues by providing basic information. After submitting the report, they will be able to communicate with HR through a thread of comments for that report.
 - i. Creating a Facility Report
 1. Title
 2. Description
 - ii. Viewing existing reports (users can only see the reports that they've submitted)
 1. Title
 2. Description
 3. Created By
 4. Timestamp
 5. Status (Open, In Progress, Closed)
 - iii. Adding or updating comments on a Facility Report
 1. Description
 - iv. Viewing a report's list of comments
 1. Description
 2. Created By
 3. Timestamp (either last modified or time created)

For HR (Angular)

1. Login page
 - a. This should be exactly the same as the employee's login page. The system should detect whether an account is for an employee or HR.
 - i. HR accounts can be hard-coded in the database.
 - ii. HR is just an employee with the HR role.
2. Navigation bar
 - a. **Home, Employee Profiles, Visa Status Management, Hiring Management, Housing Management**
3. Employee Profiles page
 - a. Allows HR to see a summary of each employee's profile, search for a particular employee, and view their entire profile.
 - b. **Summary View** (list the total number of employees, and order them alphabetically by last name)
 - i. Name (legal full name)
 1. The name should be a link that opens a new tab that displays the entire profile.
 - ii. SSN
 - iii. Work Authorization Title
 - iv. Phone Number
 - v. Email
 - c. **Search bar**
 - i. Type in the employee's first name, last name, preferred name, and it should display a matching list on every key press.
 - ii. Cover all cases for search results: one record found, multiple records found, no records found
4. Visa Status Management page
 - a. **In Progress:** Lists all employees who have not yet uploaded and been approved for all required OPT documents. It should specify what the next step is for that employee.
 - i. Ex: submitted OPT receipt: next step is to wait for HR approval
 - ii. For each employee:
 1. Name (legal full name)
 2. Work Authorization
 - a. Title
 - b. Start and end date
 - c. Number of Days Remaining
 3. Next steps
 4. Action
 - a. If the next step involves waiting for HR approval, show the uploaded document that requires approval. Allow the HR to view a preview of that document in the browser.

- i. HR should be able to Approve or Reject the document. If they reject it, they can also give feedback. This feedback should be visible to the employee when they access their visa status management page.
 - b. If the next step involves submitting the next document, show the “Send Notification” button.
 - i. This button will send an email to the employee as a reminder for their next steps.
 - b. **All:** Lists all visa-status employees. This is for looking at their uploaded documents. HR can search to filter for a particular employee.
 - i. For each employee:
 - 1. All data in section 1, except the Action category.
 - 2. All documents that were uploaded & approved.
 - a. HR should be able to download each document.
 - b. HR should be able to open a preview of the document in the browser.
 - ii. Search bar
 - 1. Type in the employee’s first name, last name, preferred name, and it should display a matching list on every key press.
 - 2. Cover all cases for search results: one record found, multiple records found, no records found
5. Hiring Management page
- a. HR should be able to generate registration tokens and review onboarding applications.
 - b. **Registration Token**
 - i. A “Generate token and send email” button that does what it says; the new employee should receive an email with a link. Clicking this link takes them to the registration page.
 - ii. The registration token should only be valid for 3 hours.
 - iii. There should be a history of email addresses that received a registration link. It should show the email address, person’s name, registration link, and status for whether this email address has been used to successfully register an account.
 - c. **Onboarding Application Review**
 - i. There should be three sections, one for each application status.
 - ii. **Pending:** HR should be able to view all submitted, pending onboarding applications. For each employee:
 - 1. Full name
 - 2. Email
 - 3. “View Application”
 - a. When clicked, open a new tab that displays the entire form.
 - b. HR should be able to Approve or Reject the application. If

they reject it, they can also give feedback on the application. This feedback should be visible to the employee when they access their rejected onboarding application.

- iii. **Rejected:** HR can view all submitted, rejected applications. Display all the same things, except that when clicking “View Application”, they do not have the option to give feedback, approve, or reject the application.
- iv. **Approved:** HR can view all submitted, approved applications. Display all the same things, except that when clicking “View Application”, they do not have the option to give feedback, approve, or reject the application.

6. Housing Management page

- a. HR should be able to view existing houses, add new houses, and delete houses.

- b. **Summary View:** for each house...

- i. Address
- ii. Landlord
 - 1. Legal full name
 - 2. Phone number
 - 3. Email
- iii. Number of employee residents
- iv. If HR clicks on the summary view, it should show additional details:
 - 1. Facility Information
 - a. Number of beds, mattresses, tables, chairs
 - b. Facility reports for that house (title, date, status)
 - i. Display 3-5 reports per page, sorted by date/time created
 - ii. Each report should display the title, description, created by, timestamp, and status (Open, In Progress, Closed)
 - iii. It should also display the list of comments (description, created by, timestamp)
 - iv. HR should be able to add or update their own comments
 - 2. Employee Information
 - a. Preferred or legal full name
 - i. The name should be a link that displays that employee profile.
 - b. Phone number
 - c. Email
 - d. Car information

- c. **Adding Houses:**

- i. Provide an address, landlord information, basic facility information

NOTE:

For any features that involve user input, invalid inputs should be appropriately handled, and users should be notified via alerts or DOM manipulation.

Please use **Material UI** for both the Employee and HR sides. We recommend starting the project in this order:

- Data modeling (entities, relationships, schemas)
- Set up the MVC-based backend structure
- To implement each page or feature, start with backend APIs and then connect it to the frontend for that page or feature.

Good Practices

1. Input validation should be handled client & server-side.
2. Users who access unsupported routes (/user/aijfojaiewoi) should be properly redirected.
3. Pay attention to user authorization.
 - a. Users should not be able to access other users' profiles, documents, etc.
 - b. Handle page navigation via directly typing in the url vs. clicking redirect links on the page.
4. Angular app file structure should be organized. Ex:
 - a. app/
 - i. components/
 1. This would contain folders that represent components used in a particular page. There may be a folder for components that are shared across pages, like a navigation bar.
 - ii. pages/
 - iii. services/
 - iv. interfaces/
 1. Make sure to **avoid** using the 'any' type as much as possible
 - v. guards/
 - vi. interceptors/
 - vii. store/
5. React app file structure should be organized. Ex:
 - a. src/
 - i. components/
 - ii. pages/
 - iii. interceptors/
 - iv. store/
6. Express app file structure should be organized. Ex:
 - a. Root folder/
 - i. index.js
 - ii. app.js
 - iii. config/
 - iv. models/

- v. controllers/
- vi. routers/
- vii. middlewares/
- viii. utils/

Submission

1. You do not need to submit anything. The “deadline” is the day of the demo on **Aug 26th, 2024 2:00 PM EST**, when you present your project.
2. For the presentation, your **slides** should cover the following topics:
 - a. Tech stack (what third-party libraries & services were used on the frontend and backend?)
 - b. Frontend architecture (file structure)
 - c. Backend architecture (file structure)
 - d. Data modeling (models, relationships)
 - e. Feature list (implemented, not implemented)
 - i. Which features did you struggle on, and specifically what part of the implementation?
 - ii. When talking about the remaining unfinished features, talk about how you would approach those implementations on a high level.
 - f. Work Contributions (who did what)
 - g. Demo Outline: (the workflow for onboarding & visa status management, from both the employee and HR perspective)
 - i. Registration (generate & send token, navigate to signup page)
 - ii. Onboarding Application
 - iii. HR Hiring Management (Emails, Onboarding Feedback, Approval, & Rejection)
 - iv. Employee Visa Management (Upload & Download)
 - v. HR Visa Management (Preview & Download)
 - Other
 - vi. Employee (Personal Information, Housing)
 - vii. HR (Employee Profile, Housing)