

Project Report

Project name: EasyHire (Streamlined Recruitment Scheduling)

Team:

Chandralekha Kommana
Gayathri Pendyala
Pranav Vishwanatham
SaiKiran Reddy Ramacharla
Sushma Deegoju

Product Description

The traditional career fair model is outdated, characterized by long queues and superficial interactions between job seekers and recruiters. This inefficiency loses time and frequently results in unsatisfactory matches. Through the creation of a digital platform that makes pre-screening calls and planned meetings possible, the initiative seeks to revolutionize this procedure. Job seekers and recruiters can conduct pre-screening talks to gauge shared interest prior to the event. In order to avoid lengthy delays and facilitate more focused conversations, if both parties are on the same page, they can book a specific time slot during the fair for a more in-depth talk. The software will provide updates and reminders in real-time, streamlining and speeding up the procedure. The initiative seeks to improve professional opportunities by displacing the disorganized first-come, first-serve paradigm with a more structured, pre-screened method.

Product functionalities

For our product, we successfully implemented all of the outlined user stories. The team worked collaboratively to ensure that each user story was thoroughly addressed, leading to a fully functional system by the end of all the sprints.

We have captured screenshots that demonstrate the functionality and features we have implemented in the system. These visuals not only validate our efforts but also offer stakeholders a glimpse into the current state of the project.

Home Page:

This is the HomePage of our application. Users can navigate to Blog, Fairs, Login and Registration pages.

The screenshot shows a web browser window for 'localhost:3000'. The header bar includes back, forward, and search icons, along with a URL field showing 'localhost:3000'. The main navigation bar at the top has a teal background with white text, containing links for 'Home', 'Blog', 'Fairs', and 'Login | Register'. Below the navigation, there is a large central image featuring two people looking at a computer monitor. The monitor displays a resume or profile page with a magnifying glass over it. To the left of the monitor, there are orange file folders and a coffee cup. To the right, there is a potted plant and a large pencil. The text 'From Hello to Hired!' is displayed above the monitor, and the tagline 'We create meaningful connections between companies and candidates.' is below it, followed by the subtext 'Whether you need a job or need to fill one, we're your trusted partner.'

User Registration Page:

The "EasyHire" webpage presents a user registration interface. It features a clear categorization, allowing users to identify as either a "Recruiter" or "Candidate" via distinct buttons. This distinction aids in tailoring the user experience.

The screenshot shows a web browser window for 'localhost:3000/register'. The header bar includes back, forward, and search icons, along with a URL field showing 'localhost:3000/register'. The main navigation bar at the top has a teal background with white text, containing links for 'Home', 'Blog', 'Fairs', and 'Login | Register'. Below the navigation, there is a large central button with a teal border and white text. The button contains the heading 'Join Us!' at the top, followed by two rectangular buttons: 'Recruiter' on the left and 'Candidate' on the right.

Registration Forms:

The "EasyHire" registration page contains several validations: mandatory fields for "First Name," "Last Name," and "Email ID" are indicated. Email format is validated for correctness, as indicated by "Email is invalid." Password strength is checked, requiring a minimum of 8 characters.

Recruiter Company Profile and validation:

The screenshot shows a web browser window for the URL `localhost:3000/register`. The title bar says "EasyHire". The main content area has a teal header "Join Us!" with two buttons: "Recruiter" and "Candidate". Below this is a section titled "Recruiter Form" containing five input fields: First Name ("Pranav"), Last Name ("Vishwanatham"), Email ID ("pranav0909@vt.edu"), and two password fields (both showing "....."). The last field is "Company Name" ("Virginia Tech"). A "Submit" button is at the bottom. At the bottom of the page are links for "About Us | Directions | Contact Us" and "Privacy Policy".

Recruiter Form

This is a detailed view of the "Recruiter Form" from the previous screenshot, showing validation errors for each field:

- First Name: The field is empty, and the error message "First name is required." is displayed below it.
- Last Name: The field is empty, and the error message "Last name is required." is displayed below it.
- Email ID: The field contains an invalid email address, and the error message "Email is invalid." is displayed below it.
- Password: The field is empty, and the error message "Password should be at least 8 characters." is displayed below it.
- Re-enter Password: This field is not shown in the screenshot but is implied by the context.
- Company Name: The field contains the value "Virginia Tech".

A "Submit" button is located at the bottom of the form.

Job Seeker Profile Creation and validation:

EasyHire

Home Blog Fairs Login | Register

Join Us!

[Recruiter](#) [Candidate](#)

Candidate Form

First Name

Last Name

Email ID

Password

Re-enter Password

Phone number

Degree

Eg: Java, Python, SQL server.....

Work Experience (if any)

LinkedIn Profile URL

No file chosen

© 2023 EasyHire, Inc

About Us | Directions | Contact Us

Privacy Policy

Candidate Form

First Name

First name is required.

Last Name

Last name is required.

Email ID

Email is invalid.

Password

Password should be at least 8 characters.

Re-enter Password

Phone number

Please enter a valid phone number.

Degree

Eg: Java, Python, SQL server.....

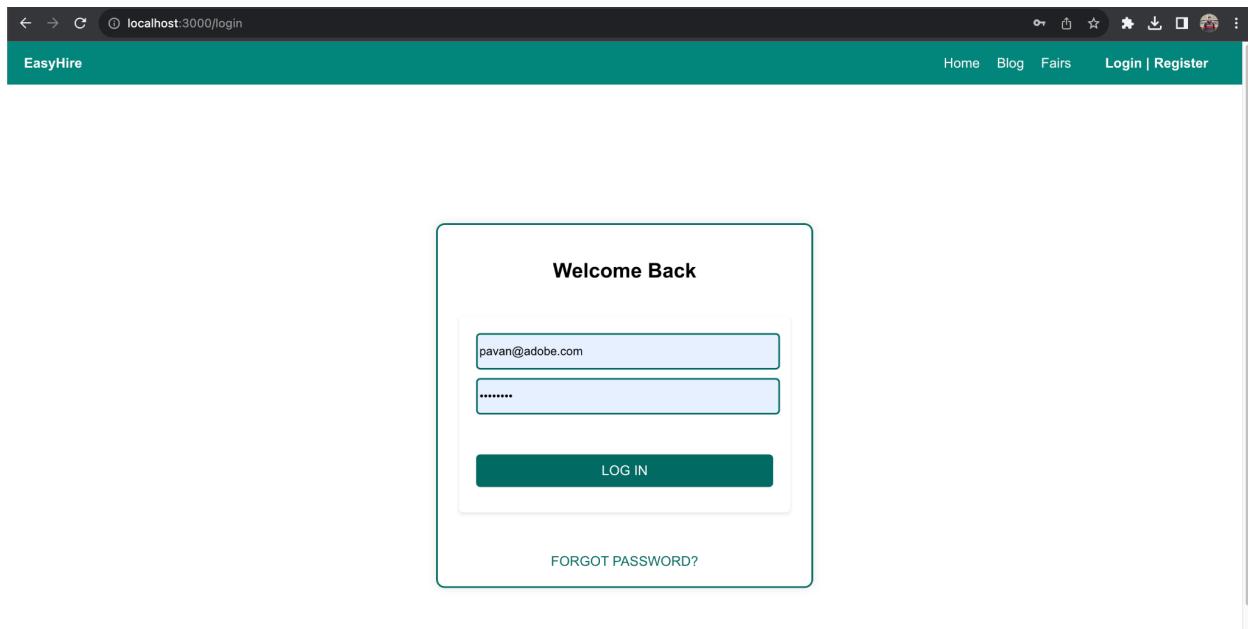
Work Experience (if any)

LinkedIn Profile URL

No file chosen

Login Page:

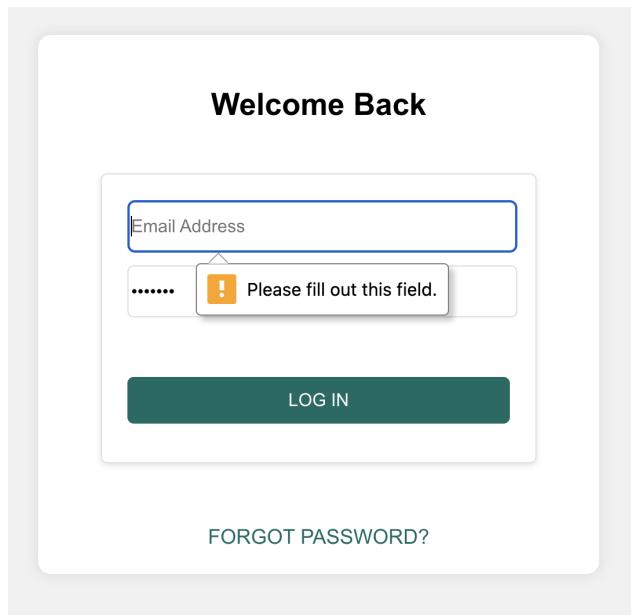
After a user successfully registers, the candidate/recruiter can login the email id and password that have been used for the registration of the user.



A screenshot of a web browser displaying the login page for 'EasyHire'. The URL in the address bar is 'localhost:3000/login'. The page has a teal header with the 'EasyHire' logo and navigation links for 'Home', 'Blog', 'Fairs', and 'Login | Register'. The main content area is titled 'Welcome Back' and contains two input fields: one for 'Email Address' containing 'pavan@adobe.com' and one for 'Password' containing '.....'. Below these fields is a teal 'LOG IN' button. At the bottom of the form is a link 'FORGOT PASSWORD?'

Login Validation:

The below screenshot shows the validation of the login page. The email id and password are mandatory fields for the user to login. The image shows the validations of the page as well.



A screenshot of the 'EasyHire' login page showing validation errors. The title 'Welcome Back' is at the top. The 'Email Address' field is highlighted with a blue border and contains 'Email Address'. The 'Password' field is empty and has a red border. A validation message 'Please fill out this field.' is displayed next to the empty password field. Below the form is a teal 'LOG IN' button and a 'FORGOT PASSWORD?' link.

Meeting Scheduling:

On the left side, there's a vertical navigation bar listing various job titles such as "Software Development Engineer", "UX Designer", "Jira Developer", etc. Each job title is associated with a company logo, indicating the employer offering the position. The central part of the screen provides detailed information about the selected job with a detailed description, key requirements, preferred qualifications, and a unique job ID. There's a section dedicated to scheduling meetings. It shows upcoming schedule times with the date and time slots and the name of the assigned recruiter. There are buttons to "Make an Appointment" for each time slot, which implies that the job seeker can book a meeting with the recruiter directly through the platform. On the page, there is a translucent notification box with the message "Your meeting is scheduled.", suggesting that the user has successfully scheduled a meeting. This feature provides immediate feedback to the user, confirming the action taken.

The screenshot shows a web application interface for 'EasyHire' at the URL localhost:3000/companies. The left sidebar lists several job openings:

- Software Developer** at **Amazon** (with Amazon logo)
- Senior Software Developer** at **Adobe** (with Adobe logo)
- Software Engineer** at **TechCorp** (with TSGI logo)
- Data Scientist** at **Data Insights Co.** (with Data Insights logo)
- UX/UI Designer** at **Creative Designs Inc.** (with Creative Designs logo)
- Software Engineer** at **Google Inc.** (with Google logo)

The main content area displays detailed information for the **Software Developer** position at **Amazon**:

Software Developer

About Us
Amazon Freight (www.freight.amazon.com) is building the technology that enables shippers to ship their own freight using the same capabilities Amazon uses to reliably deliver millions of packages every day to its customers – on time, with low cost. If you are looking for an opportunity to solve technical problems, build innovative solutions, work with top notch software developers, delight our shippers, and help build a Billion dollar business - this might be the role for you.

Key Requirements

- 3+ years of non-internship professional software development experience
- 2+ years of non-internship design or architecture (design patterns, reliability and scaling) of new and existing systems experience
- Experience programming with at least one software programming language

Preferred Qualifications

- 3+ years of full software development life cycle, including coding standards, code reviews, source control management, build processes, testing, and operations experience
- Bachelor's degree in computer science or equivalent

Job Id: AM98012

Ready to meet with a recruiter?

The screenshot shows a web-based application interface for managing company profiles and job openings. On the left, a vertical list of companies and their job roles is displayed:

- Senior Software Developer** at **Adobe** (with a red 'A' logo)
- Software Engineer** at **TechCorp** (with a blue 'TSGI' logo)
- Data Scientist** at **Data Insights Co.** (with a blue 'Data Insights' logo)
- UX/UI Designer** at **Creative Designs Inc.** (with a blue 'CD' logo)
- Software Engineer** at **Google Inc.** (with a blue 'G' logo)
- iOS Developer** at **Apple Inc.** (with a white Apple logo)
- Cloud Solutions Architect** at **Microsoft Corporation** (with a blue 'Microsoft' logo)

In the center, a specific job listing for the **Software Engineer** position at **TechCorp** is shown. It includes a message from the company: "capabilities an opportunity to help build a... ✓ Your meeting is scheduled." Below this, sections for **Key Requirements** and **Preferred Qualifications** are listed with bullet points. A note indicates the job ID is AM98012. A green button labeled "Ready to meet with a recruiter?" is present.

On the right side, two appointment scheduling entries are shown:

- Schedule Time:** Tue, Nov 24th, 2023 | 9:00 AM - 9:30 AM EDT. Assigned Recruiter: Gayathri Pandyala. A green button says "Make an Appointment".
- Schedule Time:** Tue, Nov 24th, 2023 | 12:00 PM - 12:30 PM EDT. Assigned Recruiter: Gayathri Pandyala. A green button says "Make an Appointment".

Browsing Company Profiles:

Job seekers can efficiently search and access company profiles with key information like company history and job openings. The search functionality has been implemented in the EasyHire application, enabling the system to conduct searches based on user-entered keywords. This feature efficiently retrieves and displays relevant company details corresponding to the search criteria.

The user interface of the application includes a dedicated search box, where users can enter specific keywords or company names. Upon entering a search query, the system processes this input and retrieves relevant information. The provided screenshot serves as a demonstrative example of this functionality. In this instance, the user has entered "Adobe" as the search term. Subsequently, the application effectively searches its database and displays Adobe's comprehensive company details in the search results. This showcases the application's capability to efficiently and accurately provide users with targeted company information based on their search input. Users can also search and filter with the job role. For example, if the user enters a search term 'Software Engineer', the job postings for that role are displayed.

localhost:3000/companies

EasyHire

adobe

Home Blog Fairs

Relaunch to update

Senior Software Developer

Adobe

Software Developer

About Us

Amazon Freight (www.freight.amazon.com) is building the technology that enables shippers to ship their own freight using the same capabilities Amazon uses to reliably deliver millions of packages every day to its customers – on time, with low cost. If you are looking for an opportunity to solve technical problems, build innovative solutions, work with top notch software developers, delight our shippers, and help build a Billion dollar business - this might be the role for you.

Key Requirements

- 3+ years of non-internship professional software development experience
- 2+ years of non-internship design or architecture (design patterns, reliability and scaling) of new and existing systems experience
- Experience programming with at least one software programming language

Preferred Qualifications

- 3+ years of full software development life cycle, including coding standards, code reviews, source control management, build processes, testing, and operations experience
- Bachelor's degree in computer science or equivalent

Job Id: AM98012

Ready to meet with a recruiter?

localhost:3000/companies

EasyHire

Software Engineer

Home Blog Fairs

Update

Software Engineer

TechCorp

Software Engineer

Google Inc.

Software Engineer

About Us

Join the innovative team at Google as a Software Engineer. Contribute to the development of cutting-edge technologies and work on projects that impact millions of users worldwide.

Key Requirements

- Bachelor's degree in Computer Science or related field
- 3+ years of software development experience
- Proficiency in programming languages such as Python, Java, or C++
- Strong problem-solving and algorithmic skills

Preferred Qualifications

- Master's degree in Computer Science
- Experience with distributed systems and cloud computing
- Knowledge of machine learning and artificial intelligence

Job Id: GOO12345

Ready to meet with a recruiter?

Reviewing Job Seeker Profiles:

Recruiters can navigate and assess job seeker profiles, streamlining the hiring process.

Recruiters have the ability to review candidate profiles sequentially by using the 'Next' button. They can also preview each candidate's resume directly on the page by selecting the 'View Resume' button. After reviewing, recruiters can close the resume view with an option provided. For backtracking, a 'Previous' button allows navigation to

earlier candidate information. Additionally, LinkedIn URLs are clickable, directing recruiters to the respective LinkedIn pages of candidates. All the information displayed is sourced from the candidate registration data.

The screenshot shows a web browser window titled "EasyHire" at "localhost:3000/jobSeekers". The main content area is titled "Candidate Details". It features a profile picture of a woman named Sushma. Below the picture, her details are listed: Name: Sushma, Email: sushmad@vt.edu, Phone: 5409984213, Degree: Master's in Computer Science, Experience: 2 years as SWE, and a LinkedIn Profile URL: <https://www.linkedin.com/in/sushma-deegoju-vt/>. There are two buttons at the bottom: "View Resume" and "Make an Appointment". Navigation buttons "Previous" and "Next" are also present.

The screenshot shows a modal window displaying Sushma Deegoju's LinkedIn profile. The profile includes her name, phone number (5409984213), degree (Master's in Computer Science), experience (2 years as SWE), and a LinkedIn Profile URL: <https://www.linkedin.com/in/sushma-deegoju-vt/>. A "Close" button is at the top right. At the bottom of the modal, there is a "Make an Appointment" button.

Meetings Dashboard for Recruiter:

When a recruiter logs into the EasyHire application and selects the hamburger menu, a set of options becomes available for them to navigate through the platform. These options include 'My Dashboard', 'View Candidates', and a 'Logout' feature.

The First screenshot illustrates the interface that appears when the 'View Candidates' option is selected. It displays a list of candidates, offering an organized view for the recruiter.

Additionally, there's another screenshot that shows the functionality related to scheduled meetings. In this view, recruiters can see details of their upcoming meetings, including the participant's name, the meeting location, and the scheduled time. In the same way, jobseekers can also view their scheduled meetings at one place in their meetings dashboard.

The screenshot shows a web browser window for the URL localhost:3000/jobSeekers. The page has a dark green header bar with the 'EasyHire' logo, a search bar, and navigation links for 'Home', 'Blog', and 'Fairs'. A vertical ellipsis menu icon on the right reveals a dropdown menu with four items: 'My Dashboard', 'View Candidates', and 'Logout', with 'View Candidates' being the currently selected option. The main content area is titled 'Candidate Details' and features a card for a candidate named Sushma. The card includes her profile picture, contact information (Name: Sushma, Email: sushmad@vt.edu, Phone: 5409984213), education (Degree: Master's in Computer Science), experience (Experience: 2 years as SWE), and a LinkedIn profile URL (<https://www.linkedin.com/in/sushma-deegoju-vl/>). Below the card are 'View Resume' and 'Next' buttons, and above it are 'Previous' and 'Next' buttons. At the bottom of the page, there is a footer bar with links to 'About Us', 'Directions', 'Contact Us', and 'Privacy Policy'.

Job Seeker view of dashboard:

localhost:3000/companies

EasyHire

Search jobs...

Home Blog Fairs

View Profile
View Companies
Logout

Software Developer		Software Developer	Amazon Freight (www.freight.amazon.com) is building the technology that enables shippers to ship their own freight using the same capabilities Amazon uses to reliably deliver millions of packages every day to its customers – on time, with low cost. If you are looking for an opportunity to solve technical problems, build innovative solutions, work with top notch software developers, delight our shippers, and help build a Billion dollar business - this might be the role for you.
Senior Software Developer		About Us	
Software Engineer		Key Requirements	<ul style="list-style-type: none"> • 3+ years of non-internship professional software development experience • 2+ years of non-internship design or architecture (design patterns, reliability and scaling) of new and existing systems experience • Experience programming with at least one software programming language
Data Scientist		Preferred Qualifications	<ul style="list-style-type: none"> • 3+ years of full software development life cycle, including coding standards, code reviews, source control management, build processes, testing, and operations experience • Bachelor's degree in computer science or equivalent
UX/UI Designer		Job Id: AM98012	
Software Engineer		Ready to meet with a recruiter?	

Schedule Time

Tue, Nov 24th, 2023 | 9:00 AM - 9:30 AM EDT

Assigned Recruiter

Gayathri Pendyala

Make an Appointment

Recruiter's Meetings Dashboard

localhost:3000/meetings

EasyHire

Search jobs...

Home Blog Fairs

My Meetings

View and Edit your personal meetings. [Learn More](#)

Meeting with Sofia Midkiff 12/20/2023 2:00 PM EDT / sofia09@vt.edu Venue: Squires Ballroom Booth Number: 1002	Cancel Meeting
Meeting with Sushma Deegoju 12/20/2023 12:30 PM EDT / sushmad@vt.edu Venue: Squires Ballroom Booth Number: 1002	Cancel Meeting
Meeting with Ally Sassano 12/20/2023 3:00 PM EDT / sassanoally@vt.edu Venue: Squires Ballroom Booth Number: 1002	Cancel Meeting
Meeting with Katie Brintle 12/20/2023 10:30 AM EDT / brintlek@vt.edu Venue: Squires Ballroom Booth Number: 1002	Cancel Meeting

Jobseeker's Meetings Dashboard

The screenshot shows a web browser window for 'localhost:3000/meetings'. The header includes a back arrow, forward arrow, refresh button, a search bar with placeholder 'Search jobs...', and navigation links for 'Home', 'Blog', 'Fairs', and a three-dot menu. Below the header, the title 'My Meetings' is displayed in green. A sub-instruction 'View and Edit your personal meetings. [Learn More](#)' follows. Four meeting cards are listed:

- Meeting with Chandrakala Kommana / [Amazon](#)
12/20/2023 2:00 PM EDT / chandrakala@amazon.com
Venue: Squires Ballroom Booth Number: 1001 [Cancel Meeting](#)
- Meeting with John Sable / [Google](#)
12/20/2023 12:00 PM EDT / sablejohn@google.com
Venue: Squires Ballroom Booth Number: 1006 [Cancel Meeting](#)
- Meeting with Sai Kiran Reddy Ramacharla / [Apple](#)
12/20/2023 12:30 PM EDT / sairam@apple.com
Venue: Squires Ballroom Booth Number: 1007 [Cancel Meeting](#)
- Meeting with Karen / [Adobe](#)
12/20/2024 1:30 PM EDT / karen@adobe.com
Venue: Squires Ballroom Booth Number: 1002 [Cancel Meeting](#)

© 2023 EasyHire, Inc

[About Us](#) | [Directions](#) | [Contact Us](#)

[Privacy Policy](#)

Event Host Dashboard

This is an interface for event hosts managing career fairs and recruitment events. It offers two primary views: 'Job Seekers' and 'Companies'. The 'Job Seekers' view displays a list of registered individuals, including their names and email addresses, for easy communication. The 'Companies' view shows participating companies, including their names, booth numbers, and locations, for efficient event planning. Both views are essential for effective attendee information and logistics management.

The screenshot shows a web browser window for 'localhost:3000/meetings'. The header includes a back arrow, forward arrow, refresh button, a search bar with placeholder 'Search jobs...', and navigation links for 'Home', 'Blog', 'Fairs', and a three-dot menu. Below the header, the title 'VT CS Career Fair 2023' is displayed in green. A sub-instruction 'Location: Squires CommonWealth Ballroom' follows. A dropdown menu is open, showing 'Please select an option to view' and a 'Companies' option. Below the dropdown, the section 'Companies attending:' is shown with a table:

Company Name	Booth No.
Amazon	1001
Adobe	1002
TechCorp	1003
Data Insights Co.	1004
Creative Designs Inc.	1005
Google Inc.	1006
Apple Inc.	1007
Microsoft Corporation	1008
Goldman Sachs Group Inc.	1009

EasyHire

Search jobs...

Home Blog Fairs

⋮

VT CS Career Fair 2023

Location: Squires CommonWealth Ballroom

Please select an option to view

Job Seekers

Job Seekers attending:

Name	Email
Sushma Deegoju	sushmad@vt.edu
Puneeth Kumar	puneeth@gmail.com
saikiran r	skr.ramacharla@gmail.com

© 2023 EasyHire, Inc

About Us | Directions | Contact Us

Privacy Policy

Email Notifications:

We have successfully integrated an email notification system within our platform. This feature comes into picture when a user opts to make an appointment by selecting the 'Make an Appointment' button. The system then assesses the user's available time slots and, upon selection, automatically generates and sends an email to the job seeker. This email comprehensively includes the details of the meeting, specifying both the location and the time slot that was chosen. This enhancement not only streamlines the appointment scheduling process but also ensures clear and prompt communication of vital meeting details, enhancing the overall efficiency and user experience of our platform.

EasyHire Meeting Confirmation External Inbox x

 **kirandot1976@gmail.com** 1:18 PM (0 minutes ago) ☆ ↶ ⋮
to me ▾

Dear Sushma,

Your meeting is scheduled on Tue, Nov 24th, 2023 | 10:00 AM - 10:30 AM EST with ChandraLekha Kommana from Google Inc..

Venue: Squires ballroom
BoothNumber: 1006

Looking forward to meet you!

Regards,
ChandraLekha Kommana

↶ Reply ↷ Forward

Forgot password feature

We have incorporated a 'Forgot Password' feature into our system, designed to assist users in recovering their passwords. When a user forgets their password, they can use this feature to initiate a password retrieval process. An automated email is then sent to the user's registered email address containing the user's password.

The screenshot shows an email inbox interface with various icons at the top. The main content is an email from 'kirandot1976@gmail.com' to 'to me'. The subject is 'Forgot Password Request' and it is marked as 'External'. The email body contains a message to Sushma, stating her password is 'sushma123'. It is signed off by 'Regards, EasyHire, Inc.'. At the bottom, there are 'Reply' and 'Forward' buttons.

Database: MongoDB

In the MongoDB database, various entities are structured into distinct collections. The "Companies" collection holds data about different companies, including their names, roles, and job requirements. "EventHosts" contains information about individuals hosting events, such as their names and contact details. Job seekers' details, including meeting schedules and venue information, are stored in the "JobSeekers" and "JobSeekerMeetings" collections, respectively. Similarly, the "Recruiters" collection stores recruiter details, and the "RecruiterMeetings" collection holds details about their scheduled meetings. A special "RecruiterRegistrations" collection records recruiter registrations, including company affiliations and time slots for meetings. These collections are interconnected through unique identifiers like `_id`, forming a structured and relational MongoDB database for efficient data management.

MongoDB Compass - Capstone/test.recruiters

Capstone

My Queries

Databases

Search

- admin
- local
- oplog.rs
- test
 - companies
 - eventhosts
 - jobseekermeetings
 - jobseekers
 - recruitermeetings
 - recruiters

test.recruiters

Documents Aggregations Schema Indexes Validation

Filter Type a query: { field: 'value' } or [Generate query](#)

[Explain](#) [Reset](#) [Find](#) [Options](#)

[ADD DATA](#) [EXPORT DATA](#)

1-9 of 9

```
_id: ObjectId("6563f3a8327aa01f05f83b05")
firstName: "Chandralekha"
lastName: "Kommana"
email: "kommana.chandralekha13@gmail.com"
password: "12345678"
confirmPassword: "12345678"
companyName: "Google Inc."
timeSlots: Array (3)
--v: 0

_id: ObjectId("6564185356385131cc45a15")
firstName: "Amaudip"
lastName: "Reddy"
email: "sunnyreddy738@gmail.com"
password: "12345678"
confirmPassword: "12345678"
companyName: "TechCorp"
timeSlots: Array (3)
--v: 0

_id: ObjectId("1c5641fd00e80f198f7ce844b1")
```

MongoDB Compass - EasyHire/test.jobseekers

EasyHire

My Queries

Databases

Search

- admin
- local
- test
 - companies
 - eventhosts
 - jobseekermeetings
 - jobseekers
 - recruitermeetings
 - recruiters

test.jobseekers

Documents Aggregations Schema Indexes Validation

Filter Type a query: { field: 'value' } or [Generate query](#)

[Explain](#) [Reset](#) [Find](#) [Options](#)

[ADD DATA](#) [EXPORT DATA](#)

1-4 of 4

```
_id: ObjectId("655d10619ea5a2603859f73e")
firstName: "Sushma"
lastName: "Deegoju"
email: "sushma@devt.edu"
password: "sushma123"
confirmPassword: "sushma123"
phone: "5409984213"
degree: "Master's in Computer Science"
skillset: "C++"
workExperience: "2 years as SWE"
linkedin: "https://www.linkedin.com/in/sushma-deegoju-vt/"
resume: "https://drive.google.com/file/d/1jhPg_R_dyx8HRx1ldhcdaZHQqGUgpa5/prev..."
--v: 0

_id: ObjectId("6563ec9ae4c715d6fa6e338f")
firstName: "Puneeth"
lastName: "Kumar"
email: "puneeth@gmail.com"
password: "12345678"
confirmPassword: "12345678"
phone: "5406659854"
degree: "Masters in IT"
skillset: "Python, C"
workExperience: "no prior experience"
linkedin: ""
resume: "https://drive.google.com/file/d/1jhPg_R_dyx8HRx1ldhcdaZHQqGUgpa5/prev..."
--v: 0

_id: ObjectId("656528227531fcf1a1824e43")
```

EasyHire

Documents test.eventhosts

My Queries Databases

Search

- admin
- local
- test
 - companies
 - eventhosts**
 - jobseekermeetings
 - jobseekers
 - recruitermeetings
 - recruiters

test.eventhosts

1 DOCUMENTS 1 INDEXES

Documents Aggregations Schema Indexes Validation

Filter Type a query: { field: 'value' } or Generate query + Explain Reset Find Options

ADD DATA EXPORT DATA

```
_id: ObjectId('65652c75d46f6ac0c70d9a21')
firstName: "John"
lastName: "Walker"
email: "cs@easyhire.com"
password: "eventhost"
confirmPassword: "eventhost"
__v: 0
```

1-1 of 1

> MONGOSH

EasyHire

Documents test.companies

My Queries Databases

Search

- admin
- local
- test
 - companies
 - eventhosts
 - jobseekermeetings
 - jobseekers
 - recruitermeetings
 - recruiters

test.companies

2 DOCUMENTS 1 INDEXES

Documents Aggregations Schema Indexes Validation

Filter Type a query: { field: 'value' } or Generate query + Explain Reset Find Options

ADD DATA EXPORT DATA

1-9 of 9

```
_id: ObjectId('655e1c5963c487b4e7a24247')
name: "Amazon"
role: "Software Developer"
description: "Amazon Freight (www.freight.amazon.com) is building the technology tha..."
requirements: "3+ years of non-internship professional software development experienc..."
preferred: "3+ years of full software development life cycle, including coding sta..."
jobID: "AM98012"
__v: 0
recruiter2: "John Sable"
timeslot2: "Fri, Nov 26th, 2023 | 2:00 PM - 3:00 PM EDT"
boothNum: "1001"

_id: ObjectId('655e1ca4e0c0349c7b76fb30')
name: "Adobe"
role: "Senior Software Developer"
description: "Amazon Freight (www.freight.amazon.com) is building the technology tha..."
requirements: "3+ years of non-internship professional software development experienc..."
preferred: "3+ years of full software development life cycle, including coding sta..."
jobID: "AD98012"
__v: 0
recruiter1: "Sai kiran Reddy Ramacharla"
recruiter2: "Karen Williams"
timeslot1: "Fri, Dec 9th, 2023 | 11:00 AM - 12:00 PM EDT"
timeslot2: "Thu, Jan 7th, 2024 | 09:00 AM - 10:00 AM EDT"
boothNum: "1002"

_id: ObjectId('656258e147798409a3876b7b')
name: "TechCorp"
role: "Software Engineer"
description: "TechCorp is a leading provider of software solutions for the automotive industry. We are looking for highly motivated individuals to join our team as Software Engineers. Our ideal candidate will have a strong background in software development, particularly in C++ and Python, and experience working with real-time systems. We offer a competitive compensation package, excellent benefits, and opportunities for career growth and development."
```

EDIT **CREATE** **DELETE**

> MONGOSH

EasyHire

Documents test.jobseekermeetings...

My Queries Databases

Search

admin local test

jobseekermeetings ...

Companies eventhosts jobseekers recruitermeetings recruiters

Documents Aggregations Schema Indexes Validation

Filter Type a query: { field: 'value' } or Generate query

ADD DATA EXPORT DATA

1-8 of 8 Explain Reset Find Options

_id: ObjectId('6567672a8d5603b5caf6503e')
jobSeeker: ObjectId('655d10619ea5a2603859f73e')
firstName: "Gayathri"
lastName: "Pendyala"
email: "gayathri@amazon.com"
timeSlot: "Tue, Nov 24th, 2023 | 9:00 AM - 9:30 AM EST"
venue: "Squires CommonWealth Ballroom"
boothNumber: "1001"
__v: 0
company: "Amazon"

_id: ObjectId('6567774f27f957857c99ffe7')
jobSeeker: ObjectId('655d10619ea5a2603859f73e')
firstName: "Sarah"
lastName: "Johnson"
email: "sarahapple.com"
timeSlot: "Tue, Nov 24th, 2023 | 5:00 PM - 5:30 PM EDT"
venue: "Squires CommonWealth Ballroom"
boothNumber: "1007"
company: "Apple Inc."
__v: 0

_id: ObjectId('656794a55060e62941d14e81')
jobSeeker: ObjectId('6563ec9ae4c715d6fa6e338f')
firstName: "Anudeep"
lastName: "Reddy"
email: "sunnyreddy738@gmail.com"
timeSlot: "Tue, Nov 24th, 2023 | 9:00 AM - 9:30 AM EDT"
venue: "Squires CommonWealth Ballroom"
boothNumber: "1003"
__v: 0

EasyHire

Documents test.recruitermeetings...

My Queries Databases

Search

admin local test

companies eventhosts jobseekermeetings jobseekers recruitermeetings ...

recruiters

Documents Aggregations Schema Indexes Validation

Filter Type a query: { field: 'value' } or Generate query

ADD DATA EXPORT DATA

1-14 of 14 Explain Reset Find Options

_id: ObjectId('656792c41ca4265affc39eed')
recruiter: ObjectId('656418535e6385131cc45a15')
email: "sushma@vt.edu"
timeSlot: "Tue, Nov 24th, 2023 | 10:00 AM - 10:30 AM EDT"
venue: "Squires CommonWealth Ballroom"
boothNumber: "1003"
school: "Virginia Tech"
userName: "Sushma Deegoju"
__v: 0

_id: ObjectId('656794a55060e62941d14e7f')
recruiter: ObjectId('656418535e6385131cc45a15')
userName: "Puneeth"
email: "puneeth@gmail.com"
timeSlot: "Tue, Nov 24th, 2023 | 9:00 AM - 9:30 AM EDT"
venue: "Squires CommonWealth Ballroom"
boothNumber: "1003"
school: "Virginia Tech"
__v: 0

_id: ObjectId('6567fe23e0da567732518clc')
recruiter: ObjectId('656418535e6385131cc45a15')
username: "Sushma"
email: "sushma@vt.edu"
timeSlot: "Tue, Nov 24th, 2023 | 9:00 AM - 9:30 AM EDT"
venue: "Squires CommonWealth Ballroom"
boothNumber: "1003"
school: "Virginia Tech"
__v: 0

Design

GRASP Patterns:

Five GRASP patterns for the EasyHire project:

Information Expert:

- User: The User class should hold details, such as user registration information (e.g., username, password), and user profile data.
- Profile: The Profile class should be responsible for managing and storing job seeker profiles, including personal information, employment history, skills, and uploaded documents.

Creator:

- User creates instances of Profile:

When a user creates an account, they also create a profile to showcase their skills and experience.

- UX Designer creates instances of User Interface:

The UX designer is responsible for designing the user interface for the Easy Hire application.

High Cohesion:

To ensure high cohesion, each class should have a clear, well-defined responsibility:

- The User class focuses on user-related functionalities, including registration, login, and account management.
- The Profile class is dedicated to managing job seeker profiles.
- The User Interface class focuses on creating an intuitive and visually appealing user interface.

Low Coupling:

To ensure low coupling, aim to minimize dependencies between classes:

- The User class should interact with the User Interface for user registration and login but should not have direct dependencies on other classes like Profile.
- The Profile class should interact with the User class to associate profiles with users but should not have direct dependencies on other classes.

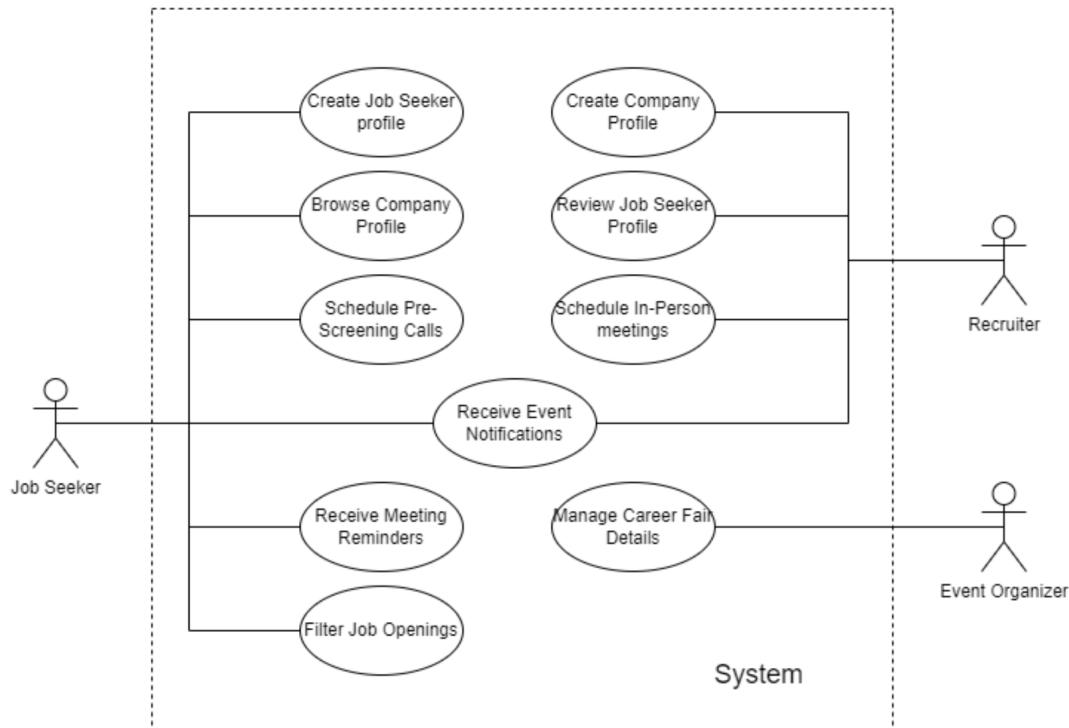
Controller:

For the controller scenario:

- There should be a controller responsible for managing user registration, login, and profile creation. This controller will coordinate interactions between the User, User Interface, and Profile classes.
- The controller should handle error cases, password reset functionality, and the flow of actions during user registration and profile creation.

These GRASP patterns help identify responsibilities and interactions between classes and guide the design of the EasyHire application, ensuring a clear and well-structured architecture.

Use Case Diagram



Actors: These are the entities that interact with the system. In this diagram, there are three actors identified:

Job Seeker: Someone looking for employment opportunities.

Recruiter: An individual or representative of an organization seeking candidates for job positions.

Event Organizer: A person who manages and organizes career-related events.

System: This represents the boundaries of the software or platform that facilitates the job seeking and recruitment process.

Use Cases: These are the actions or processes that the actors can perform with the system. Each use case is represented by an oval and is connected to the respective actor with a solid line, indicating who can perform which function. In this diagram, the use cases are as follows:

Create Job Seeker profile: A job seeker can create a profile on the platform.

Browse Company Profile: A job seeker can view company profiles.

Schedule Pre-Screening Calls: A job seeker can set up preliminary calls, possibly with recruiters.

Receive Meeting Reminders: A job seeker receives reminders about scheduled meetings.

Filter Job Openings: A job seeker can sort through available job openings.

Create Company Profile: A recruiter can create a profile for the company.

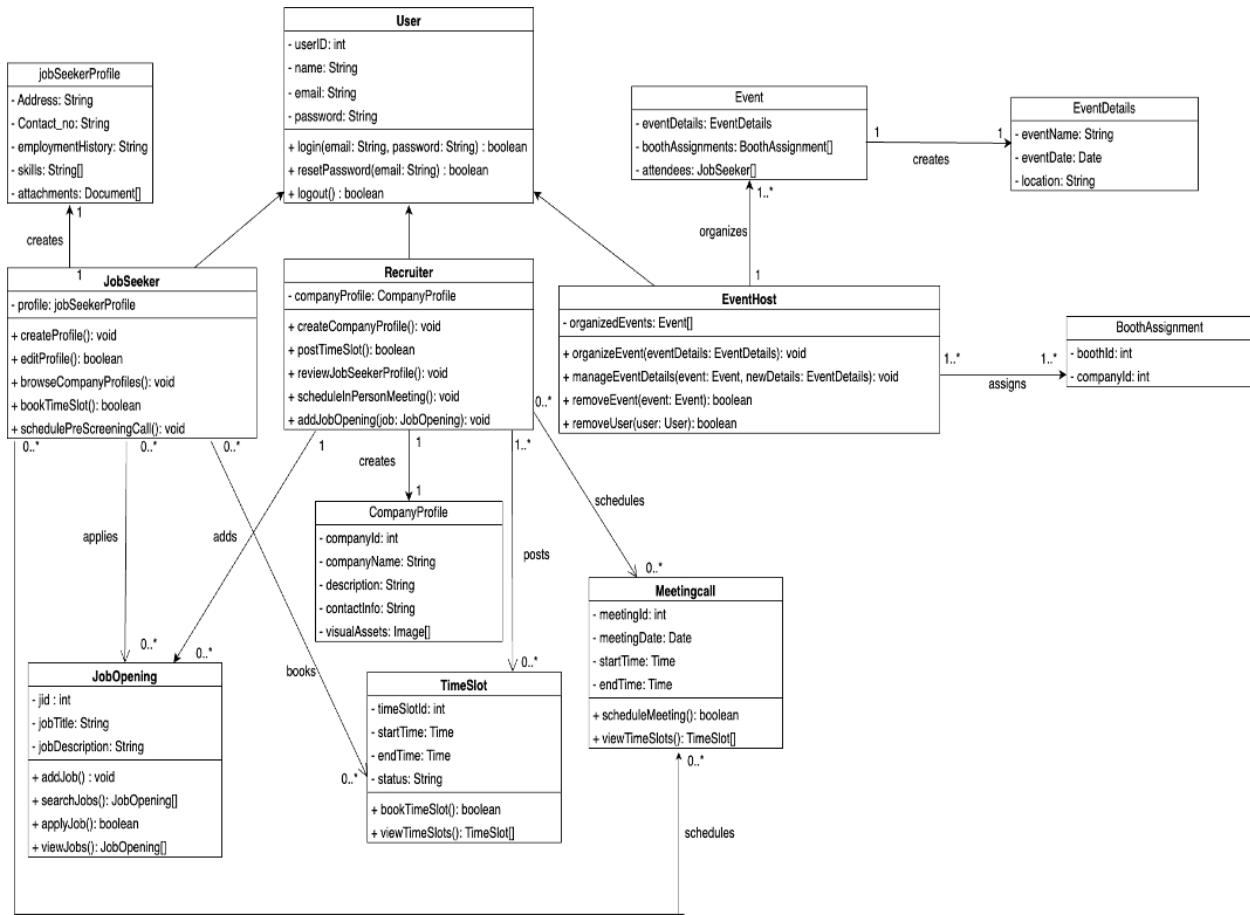
Review Job Seeker Profile: A recruiter can view profiles of job seekers.

Schedule In-Person meetings: A recruiter can arrange face-to-face meetings with job seekers.

Receive Event Notifications: A recruiter is notified of upcoming career-related events.

Manage Career Fair Details: An event organizer can handle the specifics of a career fair.

Design Class Diagram



The class diagram of EasyHire product maps out the object-oriented structure of a job recruitment platform. It outlines the system's key entities and their interactions, giving us an insight into how the application will function from a design perspective.

Main Classes and Relationships:

User: This is the foundational class for anyone who interacts with the EasyHire system, encapsulating essential attributes like identification and contact information, as well as authentication methods.

JobSeeker: Inherits from User and is tailored for individuals seeking employment. It has attributes for detailed personal and professional information, such as employment history and skills. JobSeekers can create profiles, apply to job openings, and schedule meetings with recruiters.

Recruiter: Also inheriting from User, this class is for users who represent employers. Recruiters can create company profiles, list job openings, and schedule meetings with potential candidates.

CompanyProfile: Contains detailed information about the employer, such as the company name and job roles, which helps job seekers understand more about potential employers.

JobOpening: Represents the different job listings posted by recruiters. It includes detailed job descriptions and requirements, enabling job seekers to find suitable job roles.

TimeSlot: Pertains to specific time slots available for meetings or interviews, which recruiters can book and manage.

MeetingCall: Concerns the actual scheduling of calls or meetings between job seekers and recruiters, with attributes to manage the timing and details of these interactions.

Event: Refers to job fairs or recruitment events managed within the system.

EventHost: A specialized role for users who organize and manage recruitment events. They have tools to manage event details, including the layout of booths and scheduling.

Supporting Classes:

EventDetails and **BoothAssignment**: Provide specific information about events, such as their name, date, and location, and booth assignments for companies at these events.

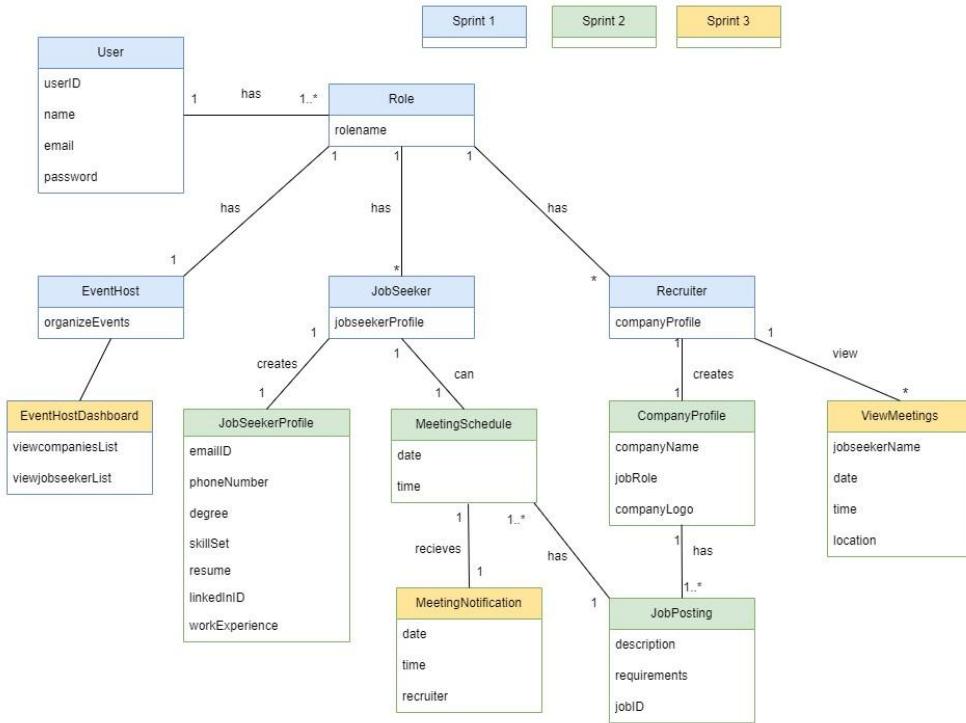
Design Considerations:

The design emphasizes modularity, with clear distinctions between different user types and their respective actions and attributes.

It supports scalability, allowing new roles or functionalities to be added as the system evolves.

It ensures data integrity by establishing clear relationships between different entities, like job openings and company profiles.

Domain Model



High-level overview of the EasyHire system's structure. It categorizes the system's main entities into classes and specifies their relationships. Here's a summarized explanation:

Core Concept: Users and Roles

User: The central entity representing all individuals who interact with the system, identified by unique attributes like ID, name, email, and password.

Role: An entity that extends from the User, defining specific roles within the system such as JobSeeker, Recruiter, and EventHost, indicating that users can have different roles with different permissions and functionalities.

Job-Seeking Functionality

JobSeeker: A user role dedicated to individuals seeking employment, with a profile that includes personal and professional details.

JobSeekerProfile: The profile associated with a JobSeeker, storing data like contact information, skills, education, and work experience.

MeetingSchedule: Represents the meetings a JobSeeker schedules, detailing when these meetings occur.

MeetingNotification: Notifications related to the MeetingSchedule, providing JobSeekers with timely information about their upcoming meetings.

Recruiting Functionality

Recruiter: A user role for individuals or entities looking to hire, associated with a CompanyProfile.

CompanyProfile: Contains information about the recruiter's company, including name, job roles, and company branding like logos.

JobPosting: Job vacancies posted by the Recruiter, detailing job descriptions and requirements.

Event Organization

EventHost: A user role for those organizing job fairs and other events, equipped with functionalities to manage event logistics.

EventHostDashboard: A specialized dashboard for EventHosts, with tools to view and manage company lists and job seeker lists for events.

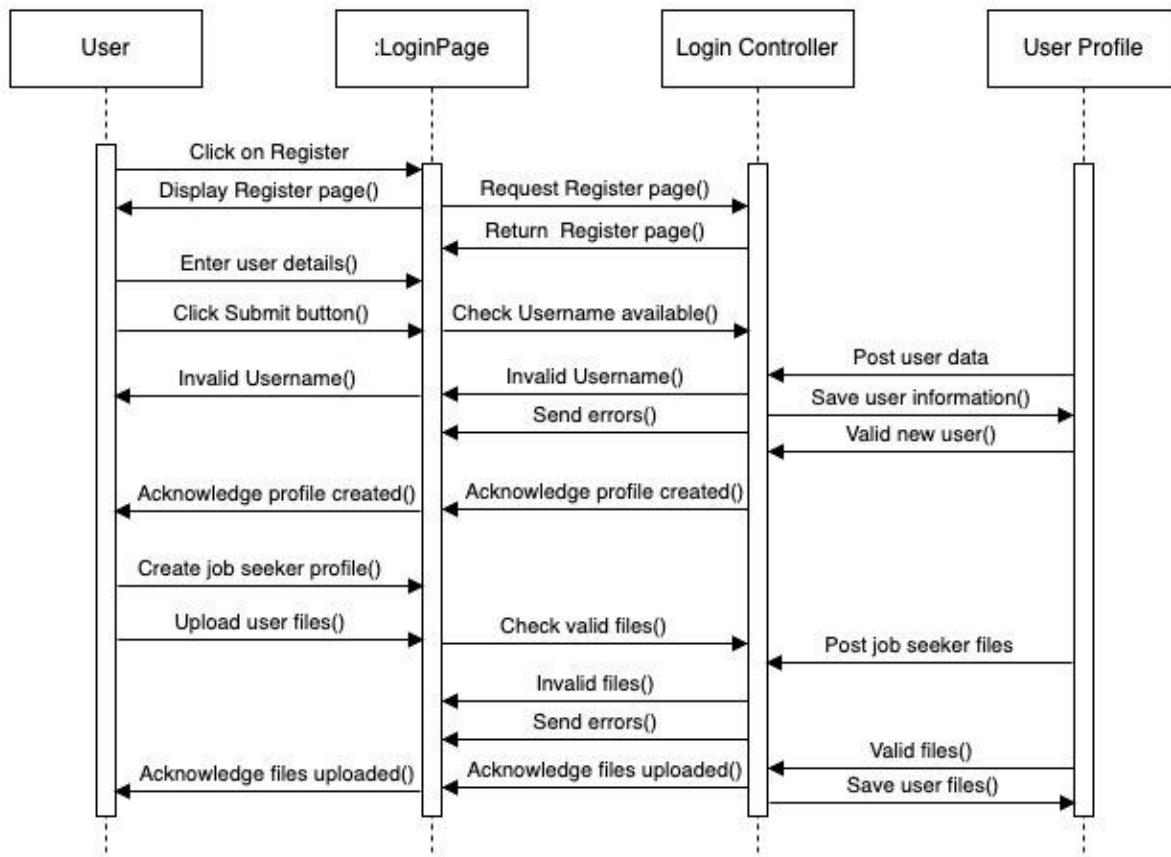
Interaction and Viewing

ViewMeetings: A class that likely allows Recruiters to view details about meetings with JobSeekers, ensuring they're prepared for upcoming interactions.

This domain model suggests a system designed with flexibility, allowing users to interact with the system according to their specific role—seeking jobs, recruiting, or hosting events. The model underpins the database design, ensuring that data is organized logically to support the system's intended functionalities.

Interaction Diagrams:

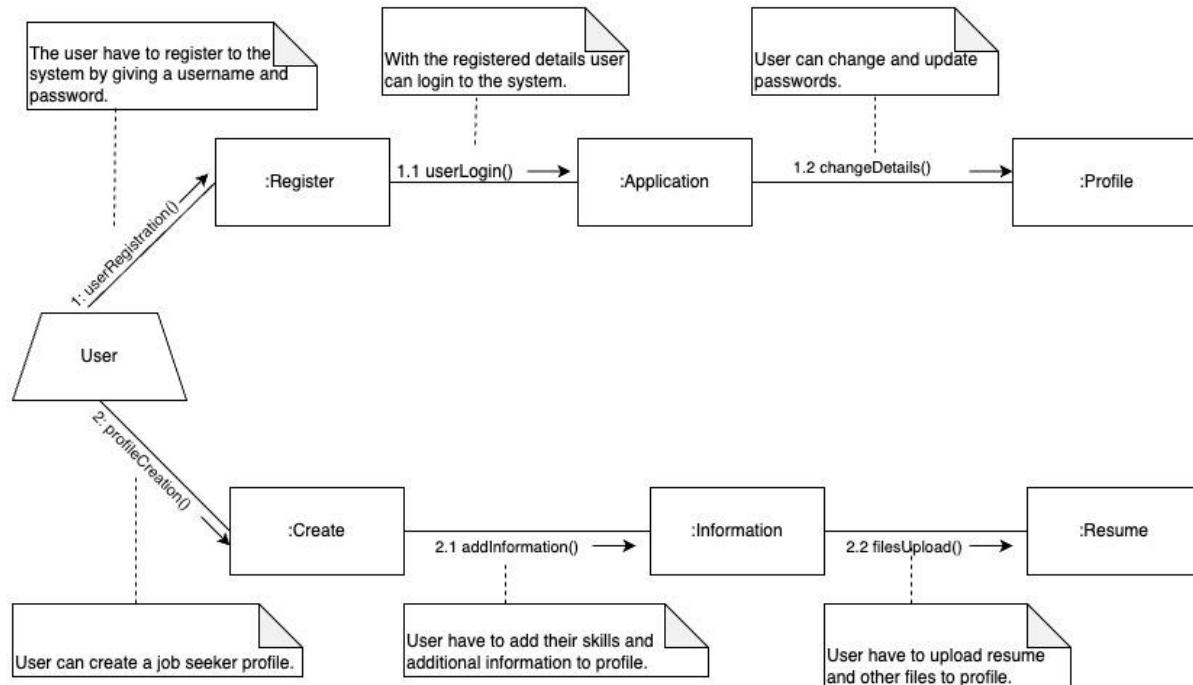
Sequence diagram- Sprint 1



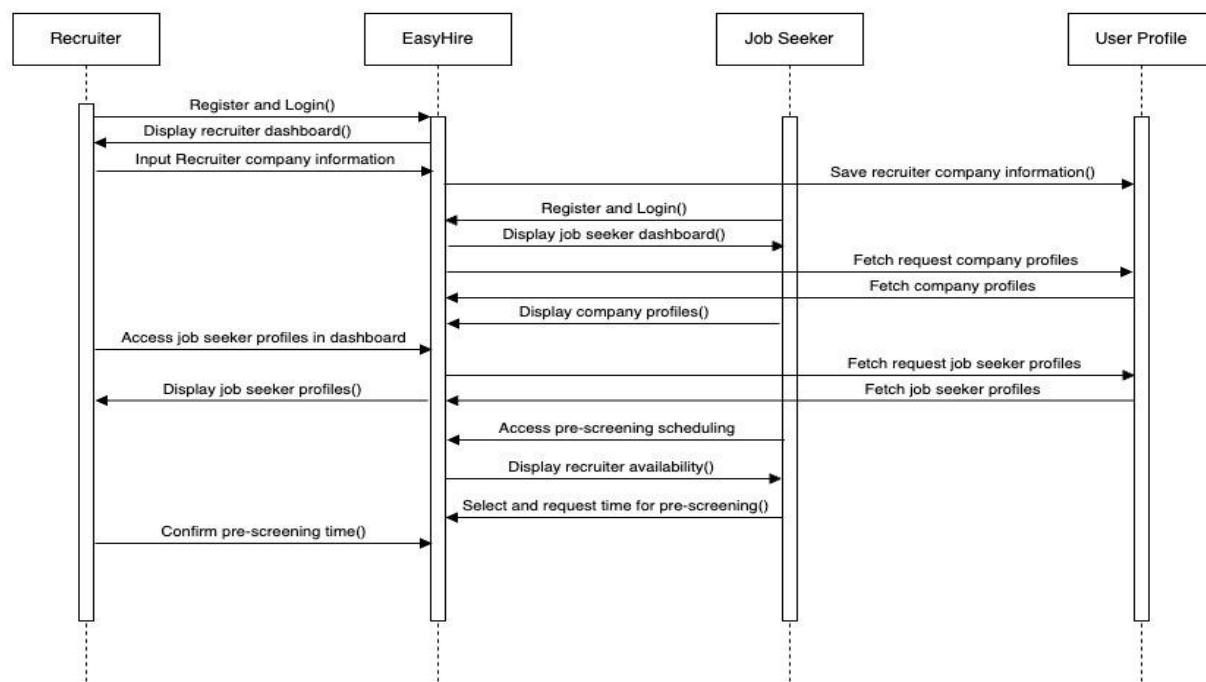
- I. When a user clicks on the register button, the register request is posted to the Web API.
- II. The response is returned and the user is asked to enter user details.
- III. When the user enters the details and clicks on the submit button, the username is validated if it already exists and if yes, an error message is displayed. Else, the user is saved in the database and new user's details are sent back to the interface.
- IV. The job seeker then enters the portal and creates their profile and uploads required files like Resume, Portfolio etc.
- V. The service validates the files and saves them to the database.
- VI. An acknowledgement is sent to the user that the upload was successful.

Collaboration Diagram - Sprint 1

A collaboration diagram depicts the associations and interactions between things. Collaboration diagrams are generally more appropriate for analysis tasks.



Sequence diagram- Sprint 2



1) Recruiter Actions:

- Register and Login: The recruiter signs up or logs into the EasyHire platform.
- Display Recruiter Dashboard: The platform displays the dashboard to the recruiter after login.
- Input Recruiter Company Information: The recruiter inputs company information into their profile.
- Access Job Seeker Profiles in Dashboard: The recruiter accesses a list of job seeker profiles from their dashboard.
- Display Search Filter Option: The platform shows the recruiter options for filtering the job seeker profiles.
- Apply Filter and Search Job Seeker Profiles: The recruiter applies filters to search through job seeker profiles.
- Display Job Seeker Profiles: The platform displays the job seeker profiles based on the applied filters.
- Send Messages to Job Seeker: The recruiter sends messages to the job seeker, possibly to inquire more about their profile or to schedule a meeting.
- Notify of Pre-Screening Request: The recruiter gets a notification from the job seeker requesting a pre-screening interview.
- Confirm Pre-Screening Time: The recruiter confirms the time for the pre-screening.

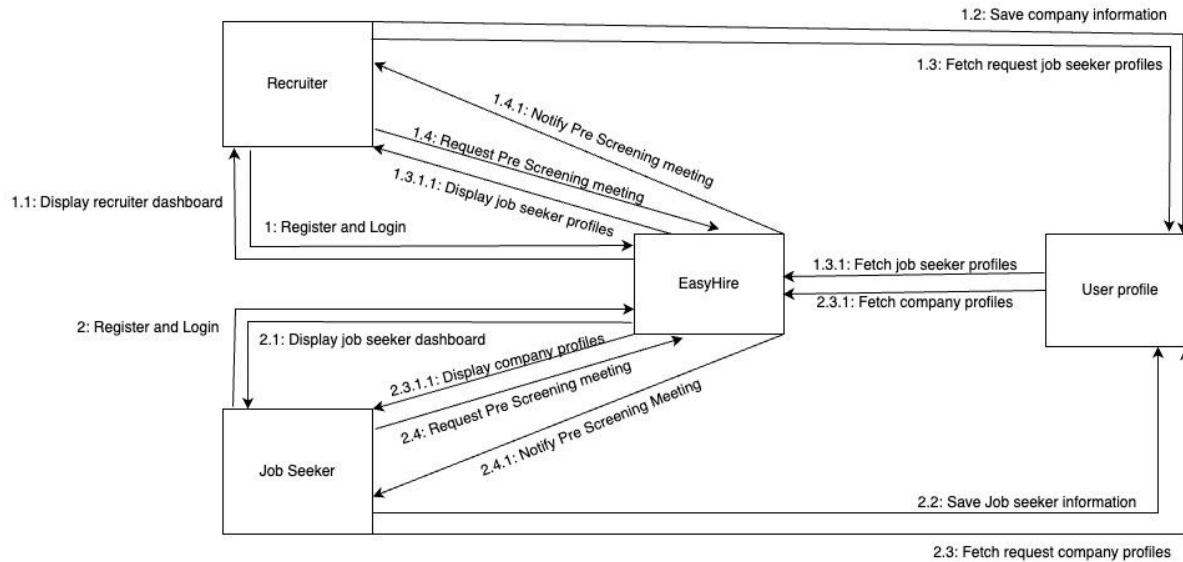
2) Job Seeker Actions:

- Register and Login: Similar to the recruiter, the job seeker signs up or logs in.
- Display Job Seeker Dashboard: The job seeker's dashboard is displayed upon successful login.
- Search and Filter Company Profiles: The job seeker uses the search and filter function to look through company profiles.
- Display Company Profiles: The platform shows the company profiles as per the job seeker's search and filter criteria.
- Access Pre-Screening Scheduling: The job seeker accesses the scheduling feature to book a pre-screening call.
- Display Recruiter Availability: The platform displays available times for the recruiter.
- Select and Request Time Slot for Pre-Screening: The job seeker selects a time slot for pre-screening and sends the request to the recruiter.

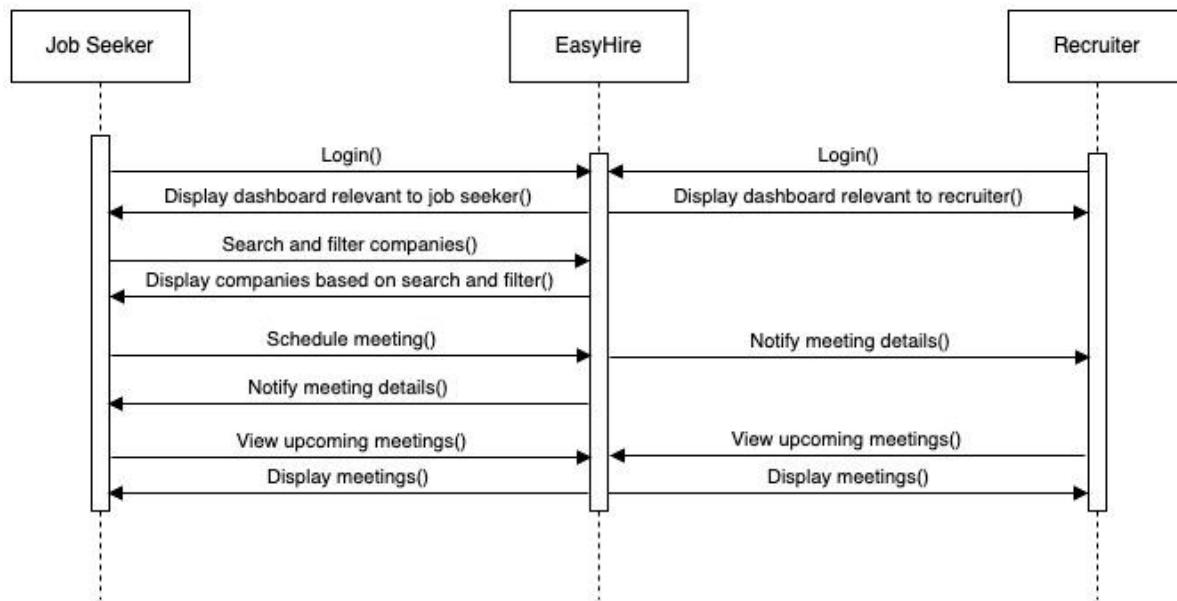
3) User Profile Actions:

- Save Recruiter Company Information: After the recruiter inputs their company information, it is saved in the user profile backend.
- Fetch Request Company Profiles: When a job seeker requests to view company profiles, the system fetches this information from the backend.
- Fetch Company Profiles: This is another instance where company profiles are fetched, perhaps as part of an update or another query.
- Fetch Request Job Seeker Profiles: When a recruiter requests to view job seeker profiles, the system fetches the profiles from the backend.
- Fetch Job Seeker Profiles: Again, job seeker profiles are fetched, potentially due to the recruiter refreshing the view or applying different filters.

Collaboration Diagram- Sprint 2



Sequence Diagram- Sprint 3



Job Seeker:

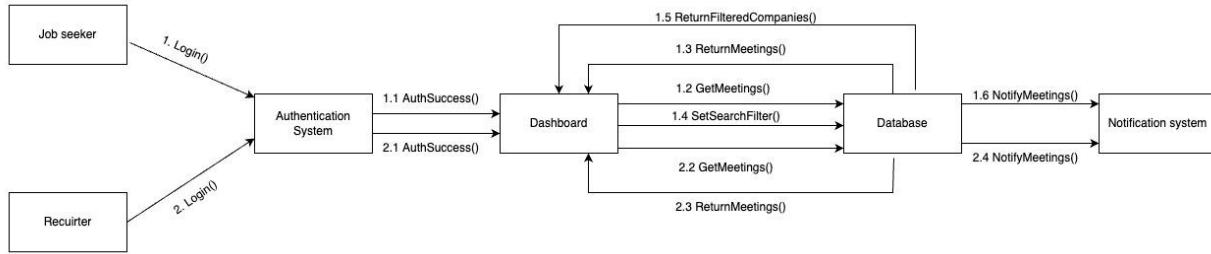
- Login(): The Job Seeker logs into the EasyHire system.
- Display dashboard relevant to job seeker(): Once logged in, the system displays the dashboard that is relevant to the Job Seeker.
- Search and filter companies(): The Job Seeker uses the system to search and apply filters to company listings.
- Display companies based on search and filter(): The system displays companies based on the search and filter criteria provided by the Job Seeker.
- Schedule meeting(): The Job Seeker can schedule a meeting using the system.
- Notify meeting details(): The system notifies the Job Seeker of the meeting details.
- View upcoming meetings(): The Job Seeker uses the system to view their upcoming meetings.
- Display meetings(): The system displays the upcoming meetings to the Job Seeker.

Recruiter:

- Login(): The Recruiter logs into the EasyHire system.
- Display dashboard relevant to recruiter(): Once logged in, the system displays the dashboard that is relevant to the Recruiter.
- Notify meeting details(): The system notifies the Recruiter of meeting details.

- View upcoming meetings(): The Recruiter uses the system to view their upcoming meetings.
- Display meetings(): The system displays the upcoming meetings to the Recruiter.

Collaboration Diagram - Sprint 3



How is it different from similar products?

EasyHire differentiates itself from other similar products in the job recruitment space through several key features and its focused approach:

1. Career Fair Orientation : Unlike general job platforms, EasyHire is specifically tailored for career fair scenarios, streamlining the experience of both job seekers and recruiters during these events.
2. Pre-Screening Calls: It facilitates pre-screening discussions between job seekers and recruiters before the actual event, ensuring that both parties can gauge interest and fit early in the process.
3. Meeting Scheduling: EasyHire allows for the scheduling of specific time slots during career fairs, which reduces wait times and ensures more meaningful, focused interactions.
4. Real-Time Notifications: The platform provides real-time updates about the event, which helps participants to be well-prepared and organized.
5. Targeted Information: For recruiters, EasyHire presents targeted and valuable information specific to career fair events, making it more focused compared to general job search platforms.

By focusing on the specific context of career fairs and providing tools and features that cater to the unique needs of these events, EasyHire provides a more structured and efficient experience compared to the broader approach of platforms like LinkedIn, Glassdoor, Indeed, and Handshake.

Tools Used

Jira

We've organized our tasks into 3 sprints using Jira. Tasks are categorized into columns, such as "TO DO", "IN PROGRESS", and "DONE". This visualization aided in quickly understanding the status of various tasks, ensuring that the team can identify bottlenecks and prioritize work efficiently.

Story points were used as a measure of the relative effort required to complete user stories or tasks. This measure is abstract and is based on the complexity, effort, and uncertainty associated with a task.

Tasks were assigned story points to understand and compare their complexities. The aggregation of these story points then provided a holistic view of the effort needed for the sprint. We used story points to gauge the total workload for each sprint. By evaluating the sum of the story points against the team's average velocity, the team ensured a balanced workload for the sprint duration. Each task in the project was assigned an estimated duration, representing the expected time to completion. As work progressed, team members updated these logs to reflect actual work hours, ensuring that deviations from estimates were tracked.

The screenshot shows the Jira software interface with the following details:

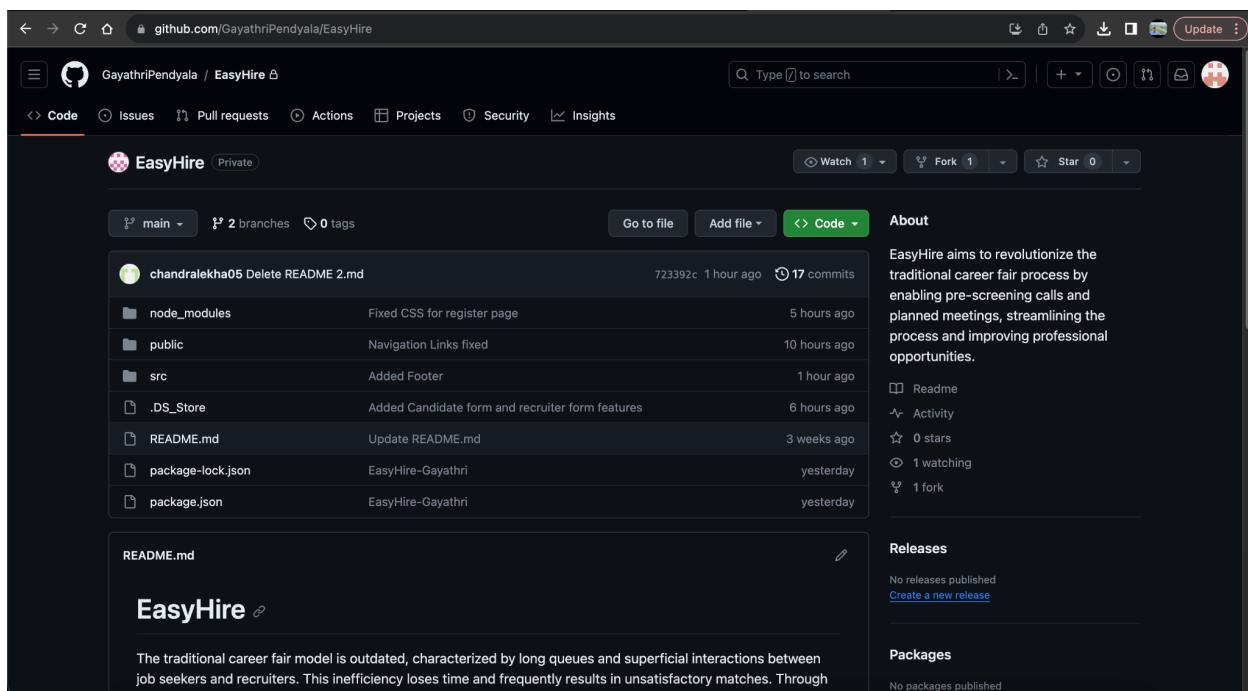
- Project Information:** Graph Masters Software project, EasyHire Sprint 1.
- Board Columns:** TO DO, IN PROGRESS, DONE.
- DONE Column Data:**
 - Develop skills section and allow to upload Resume. Story point: GM-9, Duration: 2h, Status: OK.
 - Develop interface for employer profile creation. Story point: GM-5, Duration: 2h, Status: GP.
 - Create wireframes and mockups for the application's interface. Story point: GM-3, Duration: 3h, Status: GP.
 - Develop interface for job seeker profile creation. Story point: GM-4, Duration: 2h, Status: EX.
- Sidebar Navigation:** PLANNING (Timeline, Backlog), DEVELOPMENT (Code), Project pages, Add shortcut, Project settings.
- Bottom Bar:** Quickstart button.

Git

Git helped keep track of changes, allowing all the team members to work on the same codebase without conflict, and allowing for rollback to previous states if needed.

Git facilitated collaborative effort by allowing multiple developers to work on separate branches and merge their changes when ready. By regularly committing changes, we ensured that each significant change or addition is recorded. This made it easier to understand the evolution of the project, find out when a specific change was made, and even to identify and fix problematic changes. The commit messages helped in understanding the purpose or impact of a particular commit without diving into the code changes.

The repository has multiple branches, which allowed the team to work on different features or tasks in parallel. Branching is a powerful feature of Git that allowed our team to develop features or fixes without affecting the main (or other) branches until they are ready to merge. GitHub's issue tracking helped to manage bugs, tasks, and feature requests, which help in prioritizing work and keeping track of project progress.



A screenshot of a GitHub repository page for a project named "EasyHire". The repository is private, as indicated by the "Private" badge. The page shows the following details:

- Code:** The main branch is "main", there are 2 branches, and 0 tags.
- Issues:** 0 issues.
- Pull requests:** 0 pull requests.
- Actions:** 0 actions.
- Projects:** 0 projects.
- Security:** 0 vulnerabilities.
- Insights:** Insights are available.

The repository has 17 commits from user "chandralekha05". The commits are:

File	Message	Time Ago
DELETE README 2.md	Delete README 2.md	723392c 1 hour ago
node_modules	Fixed CSS for register page	5 hours ago
public	Navigation Links fixed	10 hours ago
src	Added Footer	1 hour ago
.DS_Store	Added Candidate form and recruiter form features	6 hours ago
README.md	Update README.md	3 weeks ago
package-lock.json	EasyHire-Gayathri	yesterday
package.json	EasyHire-Gayathri	yesterday

About: EasyHire aims to revolutionize the traditional career fair process by enabling pre-screening calls and planned meetings, streamlining the process and improving professional opportunities.

Readme: Readme is available.

Activity: 0 stars, 1 watching, 1 fork.

Releases: No releases published. Create a new release.

Packages: No packages published.

Link: <https://github.com/GayathriPendyala/EasyHire>

Tech Stack & IDE

Frontend: ReactJS

For the frontend development, we chose ReactJS, a declarative and efficient JavaScript library for building user interfaces. React's component-based architecture facilitated the development of a dynamic and responsive user interface for EasyHire. This enabled us to create a seamless and interactive experience for users, with features such as:

Real-time form validation for user registration and profile management.

Dynamic scheduling components allowing users to book and manage appointments.

Responsive event dashboards, tailored for both job seekers and recruiters.

DataBase: MongoDB

MongoDB, a NoSQL database, was selected for its flexibility and performance. Its schema-less nature allowed us to iterate quickly on data models as the project requirements evolved. The database was utilized to store and manage:

User profiles, including detailed job seeker portfolios and recruiter company profiles.

Job postings and application data, enabling efficient retrieval and manipulation.

Event information and meeting schedules, ensuring robust data consistency for event management.

Backend: Node.js with Express.js Framework

In our backend development, we leveraged the power of Node.js alongside the Express framework to build a robust and scalable RESTful API. Node.js provided us with a fast and event-driven runtime, while Express streamlined the process of defining routes, handling HTTP requests, and managing middleware. This combination allowed us to create a performant backend that efficiently handled data retrieval, storage, and manipulation. The asynchronous, non-blocking nature of Node.js contributed to our API's responsiveness, making it well-suited for handling a large number of concurrent connections. With Express, we were able to implement clean and organized endpoint structures, making our API intuitive and developer-friendly. Overall, Node.js and Express proved to be a dynamic duo, empowering us to create a backend that met our project's requirements with speed and efficiency.

Development Environment

Throughout the development of EasyHire, Visual Studio Code (VS Code) served as our primary integrated development environment (IDE). Its versatile features, such as debugging, intelligent code completion, snippets, and embedded Git control, allowed for

a seamless coding experience. The lightweight nature and the extensibility of VS Code supported our agile development practices.

Retrospection

What went well and what went wrong during the process of building our product?

→ Went well

- ◆ The consistent completion of user stories across sprints indicates a deepening understanding of the project's scope and the team's capacity.
- ◆ The introduction of specific features such as email notifications and search functionality suggests an iterative approach to delivering value to the end-users.
- ◆ Positive contributions from each team member in every sprint show an engaged and dedicated team, an invaluable asset in software development.

→ Went wrong

- ◆ The team's struggle with continuous integration and collaboration hints at underlying issues such as possibly inadequate tooling, lack of experience, or insufficient automation in testing.
- ◆ The reported stress towards the sprint's end signifies a possible need for better risk management and more flexible sprint planning to accommodate unforeseen delays or complications.
- ◆ The need for improved team collaboration indicates possible gaps in communication, which may have led to inefficiencies.

What we have learned and how working on this product added to our skills?

- The project served as a practical learning experience in software development, particularly for team members with limited prior experience.
- The project facilitated firsthand experience with Agile principles, not just in theory but in a dynamic, real-world setting.
- The team learned to use issue trackers and version control systems, which are vital for professional software development.
- Learning to use Jira and Git goes beyond mere tool usage; it encompasses understanding the importance of tracking progress and managing changes in a collaborative environment.
- The experience of learning new technologies such as ReactJS and mongoDB, given the team's minimal prior experience, was enriching and will be beneficial in future projects.

- The learning curve with new technologies such as the backend development was a critical investment for the product. This experience led to more efficient development cycles ahead as the team became more proficient.
- The practice of writing unit tests and the emphasis on code quality and robustness prepares the team for maintaining larger codebases and complex systems.

What actions should have been taken to avoid the problems encountered while working on the project?

- Better sprint planning with realistic estimations could prevent work crunch and ensure more even distribution of workload and avoiding backlogs.
- Regular module integration and early prototyping would likely mitigate risks associated with late-stage integration issues.
- Active stakeholder involvement throughout the project could provide timely feedback and course corrections, avoiding misaligned product features.
- Better time management, especially allocating time for testing and avoiding end-of-sprint work crunches.
- The need for time management improvement, particularly in testing, is a common challenge in software projects. Adopting test-driven development (TDD) or behavior-driven development (BDD) could be a strategic move to ensure testing is embedded in the development process.

How could our product get better?

- Adding the capability for virtual interviews would make EasyHire a more comprehensive platform, reducing the need for external tools and streamlining the recruitment process.
- The ability to host multiple career fairs simultaneously could cater to diverse industries and time zones, increasing the platform's market penetration and also would scale its utility and appeal to a broader range of customers.
- Providing granular access and control to event hosts would allow for a customized experience, catering to the unique needs of each career fair or recruitment event.
- Adding the meeting invites to job seeker's and recruiter's calendar could be very beneficial.

In summary, the EasyHire project has demonstrated significant progress with each sprint, achieving key milestones despite initial challenges. The team's reflective practice, evidenced by their retrospectives, has fostered a culture of continuous improvement. The lessons learned from the sprint retrospectives have been instrumental in guiding the team's future actions and in shaping the trajectory of product

development. The outlined future improvements suggest a strategic vision for EasyHire, aiming to not only refine current functionalities but also to innovate and expand its offerings in the recruitment domain.