

Project Plan

Project Name:

Recruitment and HR Management System Development

Project Type:

Software Development (Agile)

Prepared by:

Akram

Date:

02/09/2024

Epics:

1. **Epic 1:** User Authentication & Role-Based Access Control (RBAC)
 2. **Epic 2:** Job Posting & Application Tracking System (ATS)
 3. **Epic 3:** Candidate Management & Interview Scheduling
 4. **Epic 4:** Employee Profile Management
 5. **Epic 5:** Leave Management System
 6. **Epic 6:** Performance Management Module
 7. **Epic 7:** Reporting and Analytics
 8. **Epic 8:** Integrations (Job Boards & Payroll System)
 9. **Epic 9:** System Administration & Security
 10. **Epic 10:** Testing, UAT & Go-Live
-

Sprint Planning:

The project is broken down into six 2-week sprints, with a timeline of approximately 12 weeks. Each sprint will focus on different epics or parts of epics, and developers, QA, and project management will be assigned accordingly.

Sprint 1: Setup & Initial Development

Duration: 2 weeks

Focus: User Authentication & RBAC + Environment Setup

- **Task 1:** Set up development environment (assigned to DevOps team)
 - **Task 2:** Implement user registration (assigned to Backend Developer)
 - **Task 3:** Develop role-based access control (assigned to Backend Developer)
-

- **Task 4:** Frontend implementation for login & signup pages (assigned to Frontend Developer)
- **Task 5:** Create basic user roles (HR Manager, Recruiter, Candidate) (assigned to Backend Developer)
- **Task 6:** Unit testing for authentication module (assigned to QA Engineer)

Team Members:

- Backend Developer: John
 - Frontend Developer: Sarah
 - DevOps: Mark
 - QA Engineer: Lisa
-

Sprint 2: Job Posting & ATS

Duration: 2 weeks

Focus: Job Posting & Application Tracking

- **Task 1:** Develop job posting module (assigned to Backend Developer)
- **Task 2:** Build job listing page and application form UI (assigned to Frontend Developer)
- **Task 3:** Enable recruiters to manage job postings (assigned to Backend Developer)
- **Task 4:** Implement application tracking system (ATS) logic (assigned to Backend Developer)
- **Task 5:** Frontend integration for ATS (assigned to Frontend Developer)
- **Task 6:** Initial testing and bug fixing (assigned to QA Engineer)

Team Members:

- Backend Developer: John
 - Frontend Developer: Sarah
 - QA Engineer: Lisa
-

Sprint 3: Candidate Management & Employee Profiles

Duration: 2 weeks

Focus: Candidate Management, Interview Scheduling, Employee Profile

- **Task 1:** Build candidate profile management feature (assigned to Backend Developer)
 - **Task 2:** Develop interview scheduling functionality (assigned to Frontend Developer)
 - **Task 3:** Develop employee profile management module (assigned to Backend Developer)
 - **Task 4:** UI for employee and candidate profiles (assigned to Frontend Developer)
 - **Task 5:** Create automated email notifications for interview scheduling (assigned to Backend Developer)
-

- **Task 6:** End-to-end testing of job application and candidate management (assigned to QA Engineer)

Team Members:

- Backend Developer: John
 - Frontend Developer: Sarah
 - QA Engineer: Lisa
-

Sprint 4: Leave Management & Performance Review

Duration: 2 weeks

Focus: Leave Management & Performance Reviews

- **Task 1:** Implement leave request module (assigned to Backend Developer)
- **Task 2:** Develop performance review system (self-assessment, peer review) (assigned to Backend Developer)
- **Task 3:** UI design for leave management and performance review (assigned to Frontend Developer)
- **Task 4:** Notifications and approvals workflow for leave requests (assigned to Backend Developer)
- **Task 5:** Testing leave and performance management modules (assigned to QA Engineer)

Team Members:

- Backend Developer: John
 - Frontend Developer: Sarah
 - QA Engineer: Lisa
-

Sprint 5: Reporting, Analytics & Integrations

Duration: 2 weeks

Focus: Reporting, Analytics, and Job Board/Payroll Integrations

- **Task 1:** Create recruitment performance dashboard (assigned to Backend Developer)
- **Task 2:** Develop leave and employee performance reports (assigned to Backend Developer)
- **Task 3:** Integration with external job boards (assigned to Backend Developer)
- **Task 4:** Payroll system integration (assigned to Backend Developer)
- **Task 5:** UI design for report generation and visualization (assigned to Frontend Developer)
- **Task 6:** Testing integration and reporting functionality (assigned to QA Engineer)

Team Members:

- Backend Developer: John
 - Frontend Developer: Sarah
 - QA Engineer: Lisa
-

Sprint 6: Final Testing, UAT & Go-Live

Duration: 2 weeks

Focus: System Testing, User Acceptance Testing (UAT), and Go-Live

- **Task 1:** Perform system-wide testing (assigned to QA Engineer)
- **Task 2:** Conduct UAT with key stakeholders (assigned to Project Manager)
- **Task 3:** Fix any final bugs or performance issues (assigned to Backend and Frontend Developers)
- **Task 4:** Deploy the system to production (assigned to DevOps)
- **Task 5:** Post-launch support and bug fixes (assigned to QA Engineer)

Team Members:

- Backend Developer: John
 - Frontend Developer: Sarah
 - QA Engineer: Lisa
 - DevOps: Mark
-

Resources & Assignments:

1. **Project Manager:** Ensures timely delivery of each sprint, manages the Jira board, and communicates with stakeholders.
 2. **Backend Developer (John):** Responsible for backend logic, integrations, and core functionality.
 3. **Frontend Developer (Sarah):** Implements all UI/UX designs and integrates the front end with backend systems.
 4. **QA Engineer (Lisa):** Conducts testing, ensures quality, and tracks bugs.
 5. **DevOps (Mark):** Responsible for setting up the environments and ensuring smooth deployment processes.
-

Timeline:

- **Sprint 1:** Week 1 – Week 2
 - **Sprint 2:** Week 3 – Week 4
 - **Sprint 3:** Week 5 – Week 6
 - **Sprint 4:** Week 7 – Week 8
 - **Sprint 5:** Week 9 – Week 10
 - **Sprint 6:** Week 11 – Week 12
-

Key Agile Practices:

1. **Daily Standups:** Each day, all team members will have a 15-minute meeting to discuss progress, blockers, and next steps.
2. **Sprint Reviews:** At the end of each sprint, there will be a sprint review meeting to demonstrate the work completed and gather feedback.
3. **Retrospectives:** After each sprint, a retrospective will be held to evaluate what went well, what didn't, and how to improve in future sprints.
4. **Backlog Grooming:** The project manager will regularly refine the product backlog, ensuring tasks are well-defined and ready for upcoming sprints.

This structure ensures the project stays on track, adheres to Agile practices, and delivers a robust recruitment and HR management system. Let me know if you'd like to modify or further customize this plan.

40

You said:

write description of each task in each sprint to help developers. They are using node js for backend and templating for front end

ChatGPT said:

ChatGPT

Here is a detailed description of each task for developers using Node.js for the backend and templating for the front end (like EJS, Handlebars, or Pug). These descriptions provide clarity on what needs to be done in each task during the respective sprints.

Sprint 1: Setup & Initial Development

Duration: 2 weeks

Focus: User Authentication & RBAC + Environment Setup

1. **Task 1:** Set up development environment (DevOps)
 - **Description:** Install and configure the project's Node.js environment. Set up the project structure (folders, package.json), install required dependencies (e.g., Express.js, database libraries like Mongoose for MongoDB), and configure environment variables using `.env`. Set up GIT repositories for source control.
2. **Task 2:** Implement user registration (Backend Developer)
 - **Description:** Create a registration endpoint using Express.js for user sign-ups. Implement user model with fields like username, email, password (hashed with bcrypt), and role (e.g., HR Manager, Recruiter, Candidate).

Ensure data is validated (e.g., email format, password strength) before storing in the database.

3. **Task 3:** Develop role-based access control (RBAC) (Backend Developer)
 - **Description:** Implement middleware for role-based access control in Node.js. Based on user roles, restrict or allow access to specific endpoints. For example, only users with "HR Manager" role can access certain admin routes. Use JWT (JSON Web Token) for authentication and role verification.
 4. **Task 4:** Frontend implementation for login & signup pages (Frontend Developer)
 - **Description:** Design and implement login and signup pages using templating engines (e.g., EJS). Create forms for user login and registration. Use Bootstrap or another CSS framework for UI styling. Connect these forms to the backend authentication routes.
 5. **Task 5:** Create basic user roles (HR Manager, Recruiter, Candidate) (Backend Developer)
 - **Description:** In the user model, define a "role" field. For now, populate the system with three primary roles: HR Manager, Recruiter, and Candidate. These roles will dictate access to specific features and pages within the application.
 6. **Task 6:** Unit testing for authentication module (QA Engineer)
 - **Description:** Write unit tests using a Node.js testing framework like Mocha or Jest to test the registration, login, and RBAC functionality. Validate that users can sign up, log in, and are granted access based on their roles.
-

Sprint 2: Job Posting & ATS

Duration: 2 weeks

Focus: Job Posting & Application Tracking System (ATS)

1. **Task 1:** Develop job posting module (Backend Developer)
 - **Description:** Create endpoints for job creation and management. Define a "Job" model in the database with fields such as title, description, location, department, salary, and status (open/closed). Ensure the job postings are tied to the recruiter responsible for them.
 2. **Task 2:** Build job listing page and application form UI (Frontend Developer)
 - **Description:** Create a page where job seekers can view all open positions. Implement filtering (e.g., by department or location) and sorting. Use a templating engine to dynamically populate the page with job postings from the database. Add an application form linked to each job posting.
 3. **Task 3:** Enable recruiters to manage job postings (Backend Developer)
 - **Description:** Create endpoints to allow recruiters to post new jobs, edit existing jobs, and close filled positions. Ensure only authenticated recruiters can access these endpoints.
 4. **Task 4:** Implement application tracking system (ATS) logic (Backend Developer)
-

- **Description:** Develop functionality for tracking job applications. When a candidate applies for a job, store their data in the ATS (e.g., resume, contact details). Create models to manage application status (e.g., under review, interviewed, rejected).
- 5. **Task 5:** Frontend integration for ATS (Frontend Developer)
 - **Description:** Create a dashboard for recruiters where they can track candidates for each job posting. Implement tables showing candidate names, application status, and action buttons to update the status (e.g., schedule interview, reject).
- 6. **Task 6:** Initial testing and bug fixing (QA Engineer)
 - **Description:** Test the job posting and ATS modules. Write end-to-end tests to ensure job postings are created, displayed, and candidates can apply. Ensure recruiters can update the status of applicants.

Sprint 3: Candidate Management & Employee Profiles

Duration: 2 weeks

Focus: Candidate Management, Interview Scheduling, Employee Profiles

1. **Task 1:** Build candidate profile management feature (Backend Developer)
 - **Description:** Extend the candidate model to store their details (e.g., name, resume, contact info, job applied for). Provide endpoints to update, view, or delete a candidate profile. Store resumes as file uploads (ensure proper file storage configurations).
2. **Task 2:** Develop interview scheduling functionality (Frontend Developer)
 - **Description:** Create a form for scheduling interviews, allowing recruiters to pick dates and times. Use a date picker component and store interview details in the database. Display scheduled interviews on recruiter and candidate dashboards.
3. **Task 3:** Develop employee profile management module (Backend Developer)
 - **Description:** Create an employee profile model in the database that stores data such as employment history, contact info, and performance records. Provide endpoints for HR to update or view employee data.
4. **Task 4:** UI for employee and candidate profiles (Frontend Developer)
 - **Description:** Implement dynamic profile pages for candidates and employees using the templating engine. Ensure user-friendly navigation, allowing HR and recruiters to view profiles and make updates.
5. **Task 5:** Create automated email notifications for interview scheduling (Backend Developer)
 - **Description:** Integrate a Node.js email service (e.g., Nodemailer) to send automated emails to candidates when interviews are scheduled, rescheduled, or canceled. Customize email templates using the templating engine.
6. **Task 6:** End-to-end testing of job application and candidate management (QA Engineer)

- **Description:** Write tests to validate the candidate management and interview scheduling process. Test that interview notifications are sent successfully and data is updated correctly across the system.
-

Sprint 4: Leave Management & Performance Review

Duration: 2 weeks

Focus: Leave Management & Performance Reviews

1. **Task 1:** Implement leave request module (Backend Developer)
 - **Description:** Develop endpoints where employees can submit leave requests. The leave request should include details like start date, end date, reason, and type of leave. Ensure the status of each request is tracked (e.g., pending, approved, rejected).
 2. **Task 2:** Develop performance review system (Backend Developer)
 - **Description:** Create a module that allows HR and managers to input performance reviews. Include functionality for self-assessments and peer reviews, with metrics for evaluation. Store these reviews in employee profiles.
 3. **Task 3:** UI design for leave management and performance review (Frontend Developer)
 - **Description:** Implement the front-end forms for submitting leave requests and displaying performance reviews. For performance reviews, include separate sections for self-assessment and manager evaluations.
 4. **Task 4:** Notifications and approvals workflow for leave requests (Backend Developer)
 - **Description:** Integrate notifications for managers when leave requests are submitted. Build an approval workflow where managers can approve or reject leave requests, and employees receive email notifications.
 5. **Task 5:** Testing leave and performance management modules (QA Engineer)
 - **Description:** Perform testing to ensure leave requests can be submitted, approved, or rejected correctly. Validate the performance review system works as expected, and data is saved and displayed properly.
-

Sprint 5: Reporting, Analytics & Integrations

Duration: 2 weeks

Focus: Reporting, Analytics, and Integrations

1. **Task 1:** Create recruitment performance dashboard (Backend Developer)
 - **Description:** Build a reporting module that allows HR to generate reports on recruitment performance. Use aggregate functions in the database to
-

calculate KPIs like number of hires, application status, and job vacancy timelines.

2. **Task 2:** Develop leave and employee performance reports (Backend Developer)
 - **Description:** Add reporting functionality to track employee leave statistics and performance reviews over time. Provide CSV export options for reports.
3. **Task 3:** Integration with external job boards (Backend Developer)
 - **Description:** Use APIs from external job boards (e.g., LinkedIn, Indeed) to automatically post job listings from the platform. Ensure data syncing between the job board and the ATS.
4. **Task 4:** Payroll system integration (Backend Developer)
 - **Description:** Integrate with the existing payroll system using the payroll provider's API. Ensure data transfer of employee details (e.g., leave balances, salary) is secure and seamless.
5. **Task 5:** UI design for report generation and visualization (Frontend Developer)
 - **Description:** Implement the front-end interface for generating reports. Create tables, charts, and graphs to visualize recruitment, leave, and performance data dynamically.
6. **Task 6:** Testing integration and reporting functionality (QA Engineer)
 - **Description:** Test the integrations with external job boards and the payroll system. Ensure reporting modules work correctly, with accurate data and export capabilities.

Sprint 6: Final Testing, UAT & Go-Live

Duration: 2 weeks

Focus: System Testing, UAT, and Go-Live

1. **Task 1:** Perform system-wide testing (QA Engineer)
 - **Description:** Conduct full system testing, focusing on integration points, data consistency, and overall performance. Use both automated and manual tests to cover all critical areas of the system.
2. **Task 2:** Conduct UAT with key stakeholders (Project Manager)
 - **Description:** Coordinate user acceptance testing (UAT) with the stakeholders. Allow them to test the system in a controlled environment and collect feedback on system usability, performance, and features.
3. **Task 3:** Fix any final bugs or performance issues (Backend and Frontend Developers)
 - **Description:** Based on feedback from UAT and system-wide tests, resolve critical bugs and performance bottlenecks. Ensure smooth functionality across all modules before final deployment.
4. **Task 4:** Deploy the system to production (DevOps)
 - **Description:** Set up the production environment, configure the database, and deploy the Node.js application. Ensure backups and monitoring tools are in place to catch potential post-deployment issues.

5. **Task 5:** Post-launch support and bug fixes (QA Engineer)

- **Description:** Provide support during the go-live period. Monitor the system for any issues and ensure all critical bugs are resolved promptly.