

Project Requirement

Onboarding Checklist Dashboard

Project Brief Document

1. Executive Summary

Project Overview

Development of a web-based onboarding checklist dashboard to streamline and track the employee onboarding process. The application will provide new hires with a centralized view of their onboarding tasks while giving HR and managers visibility into progress and completion status.

Project Goals

- Reduce onboarding administrative overhead by 40%
- Improve new hire experience and clarity during first 90 days
- Provide real-time visibility into onboarding progress
- Centralize onboarding documentation and compliance tracking
- Create a reusable template system for different roles

Success Metrics

- All new hires use the system within first week
- Average task completion time reduced by 30%
- HR reports improved tracking and reduced follow-up emails
- 90%+ new hire satisfaction score with onboarding process

2. Problem Statement

Current Challenges

- Onboarding tasks scattered across emails, spreadsheets, and verbal instructions
- No centralized tracking of task completion status
- HR spends excessive time following up on incomplete tasks
- New hires feel overwhelmed and unclear about priorities
- Inconsistent onboarding experience across departments
- Difficulty identifying bottlenecks in the onboarding process
- Lost or misplaced onboarding documents

Target Users

1. **New Hires** - Employees in their first 90 days

Project Requirement

2. **HR Team** - Administrators managing onboarding process
 3. **Hiring Managers** - Department heads overseeing new team members
 4. **IT/Facilities** - Teams responsible for setup tasks
-

3. Scope & Features

Phase 1: MVP (Weeks 1-6)

New Hire View

- Personal dashboard with assigned tasks
- Task list organized by category
- Task details (title, description, due date, priority)
- Mark tasks as complete
- Progress indicator showing overall completion percentage
- Welcome message and company overview section

HR/Admin View

- Dashboard showing all active new hires
- Overview cards with employee info and progress
- Task completion status for each employee
- Ability to view individual employee details
- Basic task template management

Task Categories

- HR & Legal (tax forms, emergency contacts, benefits)
- IT Setup (email, hardware, software access)
- Training (compliance, product, systems)
- Team Integration (introductions, meetings, buddy system)
- Workspace (desk, access cards, parking)

Phase 2: Enhanced Features (Weeks 7-10)

- Document upload functionality
- Email notifications for due dates
- Task notes and comments
- Search and filter capabilities
- Role-based task templates
- Bulk actions for administrators

Project Requirement

Phase 3: Advanced Features (Future)

- Analytics dashboard (completion trends, bottlenecks)
- Integration with HR systems (HRMS)
- Slack/Teams integration for notifications
- Mobile app version
- Automated task assignment based on triggers
- Custom workflow builder

Out of Scope (This Release)

- Payroll integration
- Performance review tracking
- Learning management system (LMS) features
- Video conferencing integration
- Multi-language support

4. User Stories

New Hire Stories

- As a new hire, I want to see all my onboarding tasks in one place so I don't miss anything important
- As a new hire, I want to know which tasks are most urgent so I can prioritize my time
- As a new hire, I want to mark tasks complete so I can track my progress
- As a new hire, I want to upload required documents so I don't need to email them separately
- As a new hire, I want to see my overall progress so I know how much is left to complete

HR Administrator Stories

- As an HR admin, I want to see all new hires and their progress so I can identify who needs follow-up
- As an HR admin, I want to create task templates for different roles so onboarding is consistent
- As an HR admin, I want to add custom tasks for specific employees so I can handle unique situations
- As an HR admin, I want to receive notifications when tasks are overdue so I can intervene
- As an HR admin, I want to view uploaded documents so I can verify compliance

Project Requirement

Manager Stories

- As a hiring manager, I want to see my new team members' progress so I can support them
- As a hiring manager, I want to add department-specific tasks so my team gets proper training
- As a manager, I want to know when onboarding is complete so I can plan full workload assignment

5. Technical Specifications

Technology Stack

Frontend

- Framework:** React 18+ with functional components and hooks
- Routing:** React Router v6
- State Management:** React Context API (Redux if complexity grows)
- UI Components:** Tailwind CSS for styling
- Icons:** Lucide React
- Forms:** React Hook Form for validation
- Date Handling:** date-fns or day.js

Backend (Recommended)

- API:** REST API (Node.js/Express) or VKMS API
- Database:** PostgreSQL or MongoDB
- File Storage:** AWS S3/On premise Server
- Authentication:** JWT tokens

Development Tools

- Version Control:** Git/GitHub/SVN
- Package Manager:** npm or yarn
- Code Quality:** ESLint, Prettier
- Testing:** Jest, React Testing Library (Phase 2)

Data Models

Employee

```
{  
  id: string,  
  firstName: string,  
  lastName: string,  
  email: string,
```

Project Requirement

```

role: string,
department: string,
startDate: date,
managerId: string,
status: "active" | "completed" | "inactive",
profilePhoto: string (url),
createdAt: timestamp,
updatedAt: timestamp
}

```

Task

```

{
  id: string, ✓
  employeeId: string, ✓
  title: string, ✓
  description: string, ✓
  category: string, ✓
  priority: "high" | "medium" | "low", ✓
  dueDate: date, ✓
  status: "pending" | "in_progress" | "completed" | "overdue", ✓
  assignedBy: string, ✓
  completedDate: date, ✓
  notes: string, ✓
  attachments: [
    {
      id: string, ✓
      fileName: string, ✓
      fileUrl: string, ✓
      uploadedAt: timestamp, ✓
    }
  ],
  createdAt: timestamp, ✓
  updatedAt: timestamp
}

```

Template

```

{
  id: string,
  name: string,
  role: string,
  department: string,
  tasks: [
    {
      title: string,
      description: string,
      category: string,
      priority: string,
      daysAfterStart: number
    }
  ],
  createdAt: timestamp,
  updatedAt: timestamp
}

```

API Endpoints (Suggested)

Authentication

- POST /api/auth/login - User login

Project Requirement

- POST /api/auth/logout - User logout
- GET /api/auth/me - Get current user

Employees

- GET /api/employees - List all employees (admin)
- GET /api/employees/:id - Get employee details
- POST /api/employees - Create new employee (admin)
- PUT /api/employees/:id - Update employee (admin)
- DELETE /api/employees/:id - Remove employee (admin)

Tasks

- GET /api/tasks?employeeId=:id - Get tasks for employee
- POST /api/tasks - Create new task (admin)
- PUT /api/tasks/:id - Update task
- DELETE /api/tasks/:id - Delete task (admin)
- PATCH /api/tasks/:id/complete - Mark task complete
- POST /api/tasks/:id/upload - Upload attachment

Templates

- GET /api/templates - List all templates (admin)
- GET /api/templates/:id - Get template details (admin)
- POST /api/templates - Create template (admin)
- PUT /api/templates/:id - Update template (admin)
- DELETE /api/templates/:id - Delete template (admin)

Security Requirements

- Role-based access control (new hire, manager, admin)
- Employees can only view their own data
- Secure file upload with type/size validation
- HTTPS for all communications
- Password hashing and secure authentication
- Input validation and sanitization
- Protection against XSS and SQL injection

6. Design & User Experience

Design Principles

Project Requirement

- Clean, intuitive interface requiring minimal training
- Mobile-responsive design for on-the-go access
- Consistent color scheme aligned with company branding
- Clear visual hierarchy and progress indicators
- Accessible (WCAG 2.1 Level AA compliance)

Key Screens

1. New Hire Dashboard

- Welcome banner with employee name and start date
- Progress circle showing completion percentage
- Task list with filters (All, Pending, Completed)
- Category tabs for task organization
- Quick stats (tasks remaining, upcoming due dates)

2. Task Detail View

- Full task description
- Due date and priority indicator
- Complete/incomplete toggle
- Notes section
- Document upload area
- Related tasks or dependencies

3. HR Admin Dashboard

- Overview cards for each new hire
- Grid or list view toggle
- Search and filter options
- Quick actions (add task, send reminder)
- Summary statistics

4. Template Management

- List of existing templates by role
- Template editor with drag-and-drop task ordering
- Preview mode
- Duplicate template functionality

Color Scheme (Suggested)

- Primary: Blue (#2563EB) - Trust, professionalism
- Success: Green (#10B981) - Completed tasks

Project Requirement

- Warning: Amber (#F59E0B) - Due soon
- Danger: Red (#EF4444) - Overdue
- Neutral: Gray scale for backgrounds and text

7. Project Timeline

Week 1: Planning & Setup

- Requirements gathering with HR team
- Interview 2-3 recent hires for feedback
- Set up development environment
- Create wireframes and mockups
- Define data structure
- Initialize React project with folder structure

Week 2: Core Infrastructure

- Set up authentication system
- Create database schema
- Build API endpoints (or set up Firebase)
- Implement routing structure
- Create reusable UI components (buttons, cards, forms)

Week 3: New Hire View

- Build employee dashboard layout
- Implement task list component
- Create task detail view
- Add task completion functionality
- Implement progress tracking

Week 4: Admin View

- Build HR dashboard
- Create employee overview cards
- Implement task management (add, edit, delete)
- Add search and filter functionality

Week 5: Polish & Integration

- File upload implementation
- Form validation

Project Requirement

- Error handling
- Responsive design adjustments
- Integration testing

Week 6: Testing & Deployment

- User acceptance testing with HR
- Bug fixes and refinements
- Performance optimization
- Documentation
- Production deployment

Week 7-8: Phase 2 (Optional)

- Email notifications – Credentials will be shared
- Template management
- Enhanced filtering
- Analytics dashboard

8. Deliverables

Code Deliverables

- Complete React application source code
- API backend code (if applicable)
- Database migration scripts
- Environment configuration files
- README with setup instructions

Documentation

- Technical documentation
- API documentation
- User guide for new hires
- Admin guide for HR team
- Deployment guide

Design Assets

- Wireframes/mockups
- Component library/style guide
- Logo and branding assets

Project Requirement

9. Testing Strategy

Testing Levels

Unit Testing

- Component rendering tests
- Function logic tests
- Form validation tests
- Minimum 70% code coverage

Integration Testing

- API endpoint testing
- Database operations
- Authentication flow
- File upload functionality

User Acceptance Testing

- Test with 2-3 HR team members
- Test with 1-2 recent hires
- Gather feedback and iterate
- Create test scenarios checklist

Test Scenarios

1. New hire logs in and sees personalized dashboard
2. New hire marks task as complete
3. New hire uploads document
4. HR admin creates new employee profile
5. HR admin assigns tasks to employee
6. System calculates progress correctly
7. Overdue tasks are flagged appropriately
8. Search and filter return correct results
9. Mobile view displays correctly
10. Invalid inputs are rejected with clear messages

10. Risks & Mitigation

Technical Risks

Project Requirement

Risk	Impact	Probability	Mitigation
Performance issues with large datasets	High	Medium	Implement pagination, lazy loading
Browser compatibility issues	Medium	Low	Test on multiple browsers early
Security vulnerabilities	High	Medium	Regular security audits, input validation

Project Risks

Risk	Impact	Probability	Mitigation
Scope creep	High	High	Stick to phased approach, clear requirements
Delayed feedback from stakeholders	Medium	Medium	Schedule regular check-ins
Intern knowledge gaps	Medium	Medium	Provide mentorship, learning resources
Changing requirements	Medium	Medium	Build flexibility into architecture

11. Success Criteria

Technical Success

- ✓ Application loads in under 3 seconds
- ✓ Zero critical bugs in production
- ✓ 95%+ uptime
- ✓ Works on Chrome, Firefox, Safari, Edge
- ✓ Responsive on desktop, tablet, and mobile
- ✓ Code passes linting standards
- ✓ All core features functional

Business Success

- ✓ HR team adoption within 2 weeks

Project Requirement

- All new hires onboarded through system
- Positive feedback from users (4/5 rating minimum)
- Measurable reduction in HR admin time
- Complete task tracking for all employees

12. Post-Launch Plan

Maintenance

- Weekly bug triage and fixes
- Monthly dependency updates
- Quarterly security audits
- Performance monitoring

Future Enhancements

- Collect user feedback for 30 days post-launch
- Prioritize Phase 2 features based on feedback
- Plan integration with existing company systems
- Explore mobile app development
- Consider AI features (task recommendations, chatbot support)

Knowledge Transfer

- Document all code and architecture decisions
- Create video walkthrough of codebase
- Pair programming sessions with other developers
- Handoff meeting with maintenance team

13. Resources & Support

Intern Support

- **Technical Mentor:** Chetan Gholekar
- **Product Owner:** Tanishqa Ghude
- **Weekly Check-ins:** Every Monday 10 AM
- **Daily Standups:** (Optional) 15 minutes

Learning Resources

- React Official Documentation

Project Requirement

- Tailwind CSS Documentation
- MDN Web Docs
- Stack Overflow
- Company coding standards document

Tools & Access

- GitHub repository access
- Development environment setup
- Staging server access
- Design tool access (Figma/Sketch)
- Communication channels (Slack/Teams)

14. Approval & Sign-Off

Stakeholder Approval

- [] HR Director
- [] IT Manager
- [] CTO/Engineering Lead
- [] Intern's Direct Manager

Project Kick-off Date: _____

Target Launch Date: _____

Appendix A: Glossary

- **MVP:** Minimum Viable Product - the simplest version that delivers value
- **CRUD:** Create, Read, Update, Delete - basic database operations
- **API:** Application Programming Interface - how frontend and backend communicate
- **JWT:** JSON Web Token - secure authentication method
- **WCAG:** Web Content Accessibility Guidelines
- **UAT:** User Acceptance Testing

Appendix B: Reference Links

- Project Repository: [To be created]
- Design Mockups: [To be added]
- API Documentation: [To be created]

Project Requirement

- Staging Environment: [To be set up]
- Production URL: [To be deployed]

Document Version: 1.0

Last Updated: December 22, 2025

Document Owner: Chetan Gholekar Sr. Tech Lead

Next Review Date:

