BLACKFLAG HR — FULL PROJECT REPORT

Team: Prompt Pirates — Fall 2025

Members: Dev Chandralal Mulchandani, Lordphone Wen, Mikkilineni Sasi Nikhil, Nikhil Sai Venkat Manam

ABSTRACT

BlackFlag HR is a secure, cloud-native Human Resources platform designed to demonstrate how modern enterprises build HR systems using cloud infrastructure, distributed architecture, identity-based security, and DevOps best practices. The system incorporates AWS-managed services (CloudFront, WAF, ALB, VPC, ECS Fargate, RDS PostgreSQL, Secrets Manager, IAM), a FastAPI backend, a React frontend, Terraform Infrastructure as Code, and CI/CD automation using GitHub Actions. It focuses heavily on enterprise security principles, including TLS encryption, Role-Based Access Control (RBAC), private networking, OIDC-ready Single Sign-On, and secret rotation. This report presents the project's motivation, architecture, design decisions, implementation process, testing strategy, results, and future directions.

TABLE OF CONTENTS

1. INTRODUCTION

1.1 Project Goals and Objectives

The primary goal of BlackFlag HR is to build a secure, scalable HR portal that demonstrates enterprise-level architecture and security practices. Key objectives include:

- Implementing cloud-native design using AWS-managed services
- Supporting RBAC (Role-Based Access Control)
- Designing OIDC-ready authentication for Single Sign-On
- Enforcing TLS-only communication
- Deploying the system using Infrastructure as Code (Terraform)
- Automating build, test, and deployment using GitHub Actions
- Creating a modular HR system with employee management, leave features, and document workflows

1.2 Problem and Motivation

Traditional HR systems often suffer from outdated security, limited scalability, manual processes, and poor integration with modern identity providers. Enterprises today require:

- Centralized identity management
- Unified access control
- High availability
- Automated deployments
- Secure data handling

BlackFlag HR addresses these needs by implementing cloud-native architecture, identity-first design, and defense-in-depth security.

1.3 Application and Impact

BlackFlag HR can support HR operations such as employee information management, leave processing, document storage, and analytics. Academically, the project teaches students real enterprise software development practices. Industrially, the system models how companies build large-scale HR platforms for distributed workforces.

1.4 Project Results and Deliverables

The final deliverables include:

- Full HR web application (frontend + backend)
- AWS infrastructure (ECS, ALB, RDS, CloudFront, WAF)
- Terraform IaC scripts
- GitHub Actions CI/CD pipelines
- Documentation: architecture, design, deployment, testing, and this report
- Presentation slides and demo

1.5 Market Research

Major HR systems—Workday, SAP SuccessFactors, ADP, BambooHR—use cloud-native architecture, SSO, encrypted workflows, RBAC, and analytics. BlackFlag HR incorporates similar principles, providing a scaled-down but realistic enterprise model.

1.6 Project Report Structure

This report includes background, requirements, design, implementation, testing, and future work.

2. BACKGROUND AND RELATED WORK

2.1 Background and Used Technologies

BlackFlag HR uses:

- React + TypeScript for a responsive frontend
- FastAPI for a fast, typed backend
- PostgreSQL (RDS) for relational data
- AWS ECS Fargate for serverless container hosting
- AWS CloudFront + WAF + ALB for delivery and security
- AWS VPC for private networking
- AWS Secrets Manager for secret storage
- Terraform for infrastructure provisioning
- GitHub Actions for CI/CD

2.2 State-of-the-Art Technologies

Modern HR systems rely on:

- Microservices
- Identity federation (OIDC/SAML)
- Serverless compute
- Encrypted storage and transport
- Cloud-managed databases
- Automated DevSecOps pipelines

BlackFlag HR demonstrates these trends in an academic, scalable environment.

2.3 Literature Survey

Research supports identity-first systems, zero-trust networking, container deployment, IaC practices, and RBAC enforcement. HR data privacy also demands encryption and auditability. Modern DevOps practices and cloud security literature reinforce the project’s architecture.

3. SYSTEM REQUIREMENTS AND ANALYSIS

3.1 Domain and Business Requirements

The system must:

- Manage employee records securely
- Provide role-based access
- Support self-service workflows
- Keep sensitive data private
- Ensure identity-driven authentication
- Remain scalable and maintainable

3.2 Customer-Oriented Requirements

Users:

- Employees: view/update personal info, submit leave
- Managers: view team, approve leave
- HR Admins: full CRUD abilities

3.3 System Functional Requirements

- Login (OIDC-ready)
- Employee directory
- Employee CRUD (HR Admin)
- Profile management (self-service)
- Leave requests & approvals
- Document storage (planned)

3.4 System Behavior Requirements

State behaviors:

- Login → Authenticated session
- Leave request → pending → approved/declined
- Employee onboarding/offboarding lifecycle

3.5 Non-Functional Requirements

- Security: TLS-only, WAF, IAM least privilege
- Performance: <300ms p50 read latency
- Scalability: container autoscaling
- Availability: multi-AZ infrastructure
- Maintainability: Terraform, modular code

3.6 Context and Interface Requirements

- Frontend ↔ Backend through REST APIs
- Backend ↔ Database through private networking
- OIDC identity provider (Azure AD/AWS IAM Identity Center)

3.7 Technology and Resource Requirements

Requires AWS account, Docker, Terraform, Python, Node.js, GitHub Actions, and role permissions.

4. SYSTEM DESIGN

4.1 System Architecture Design

Architecture flow:
CloudFront → WAF → ALB → ECS Fargate → RDS PostgreSQL
Components reside in private subnets; only ALB is public.

4.2 System Data and Database Design

Tables:

- Employee
- User
- Roles
- Documents
- Leave Requests
- Audit Logs

Relationships: employee–manager, user–roles, employee–documents.

4.3 Interface and Connectivity Design

- API endpoints enforce RBAC
- IAM roles secure AWS interactions
- CloudFront delivers frontend
- ALB routes to ECS containers

4.4 User Interface Design

The UI contains:

- Dashboard
- Directory
- Profile page
- Leave system
- Admin panel

4.5 API and Logic Design

APIs implement:

- Auth
- CRUD operations
- Leave workflow
- Document upload (presigned URLs)

4.6 Design Decisions and Patterns

- Serverless containers > VMs
- OIDC identity > traditional login
- RBAC enforced both frontend & backend
- IaC for consistency
- Multi-layer defense security model

5. SYSTEM IMPLEMENTATION

5.1 Implementation Summary

- Frontend built in React/TypeScript
- Backend built in FastAPI
- Infrastructure automated via Terraform
- Deployed on ECS Fargate

5.2 Implementation Issues & Resolutions

- IAM permission errors → fixed via role scoping
- VPC routing mistakes → corrected subnet associations
- ECS task failures → fixed container build and health checks
- CORS errors → updated API gateway rules

5.3 Tools and Technologies Used

- React, TypeScript, Tailwind
- FastAPI, Python
- Docker
- Terraform
- GitHub Actions
- RDS PostgreSQL

6. SYSTEM TESTING & EXPERIMENTS

6.1 Testing Scope

Includes:

- Functional
- Integration
- Security
- API
- RBAC
- Load testing

6.2 Testing Approaches

Used:

- Manual UI tests
- Postman API tests
- Pytest backend tests
- AWS connectivity debugging
- TLS certificate validation

6.3 Results

- RBAC validated correctly
- Login/session flow stable
- Employee CRUD functional
- Leave flow working end-to-end
- ECS + ALB routing stable

7. CONCLUSION & FUTURE WORK

7.1 Project Summary

BlackFlag HR demonstrates secure HR workflows, enterprise-level cloud architecture, modern identity design, automated deployments, and scalable infrastructure.

7.2 Future Work

- Full Azure AD SSO integration
- Document management with presigned URLs
- HR analytics dashboards
- Performance review module
- Multi-region deployment
- Automated compliance logging

REFERENCES

- AWS Documentation
- FastAPI Documentation
- React Documentation
- Terraform Registry
- Workday/SAP HR whitepapers
- Team project design documents (DESIGN.md, FEATURES.md, ARCHITECTURE.md, DEPLOYMENT.md)