

# TIMESHEET PROJECT — 1

ADVISOR: PROFESSOR KENNETH FLETCHER  
CLIENT: BARET NWOGBU. CEO AT RYBEN STAFFING

Team:

Aaditya Tamrakar

Aditya Ambati

Uma Mahesh Aari



# PROBLEM STATEMENT

Provision of temporary staffing services to an array of industries is what we specialize in. We fill in temporary gaps in organizations workforce which helps to ensure that there is no disruption to the services the organization renders to their customers.

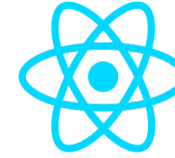
The objective of this project is to develop a web-based program that allows people to sign in and sign out of different events. This program should be able to develop reports based on different criteria like location, date range, time range, events name etc. This program will have different user levels like Admin, Manager, supervisor etc.



Backend



AWS Elastic Beanstalk



React Framework



Amazon Simple Storage  
Service (S3)

Frontend



Postman

Testing



Integration Service  
Environments

# SOFTWARE ARCHITECTURE

# SOFTWARE DEMO — DEPLOYED IN AWS FREE TIER

## **Backend API:**

<http://timesheettesting-env.eba-fakakc8k.us-east-1.elasticbeanstalk.com/>

**Frontend:** <http://dev-timesheet.s3-website.us-east-2.amazonaws.com/>

**Test Credentials:** Username: [cs682@ryben.io](mailto:cs682@ryben.io) / Password: 123

## **Postman API Documentation:**

<https://winter-meteor-947358.postman.co/workspace/timesheet-local~7906aa12-de42-4678-970f-149434218493/overview>

**Github Repository:** [https://github.com/Rybenben/Timesheet\\_project](https://github.com/Rybenben/Timesheet_project)

**Project Documentation:** [Scrapbook](#), [Documentation](#)

# LEARNING

---

AWS Beanstalk - Java Spring Boot

---

AWS S3 – Automated Deployment, React Framework

---

PostgreSQL – AWS RDS (DynamoDB\*)

---

AWS Lambda\*

---

Connectivity using Fetch API (frontend – backend)

---

Testing – Integration, Unit Testing using Postman tool

---

Troubleshooting

---

\*Note: Services tested but not implemented

# RE-DESIGN IDEAS

1. Offline App/Data sync
2. Use Realtime database (Firestore/InfluxDB) instead of AWS RDS (PostgreSQL)
3. Dashboard Calendar can be replaced with Map with geolocations tagged with Events
4. Improve backend access control and JWT authentication
5. Sign in/out using social media or OAuth Application (Google, Facebook login)
6. Better implementation of continuous integration and continuous deployment