XYZ-Portal

Software Design Document

Prepared by: Daryl Lau

Date created: 20 June 2020

Contents

[Table of Figures 3](#_Toc43596963)

[1. Introduction 4](#_Toc43596964)

[1.1 Purpose 4](#_Toc43596965)

[1.2 Scope 4](#_Toc43596966)

[1.3 Overview 4](#_Toc43596967)

[2. System Architecture 4](#_Toc43596968)

[2.1 Architectural Design 4](#_Toc43596969)

[3. Functional Requirements 5](#_Toc43596970)

[3.1 Application – Form filing 5](#_Toc43596971)

[3.2 Dashboard 7](#_Toc43596972)

[3.3 Admin Portal 8](#_Toc43596973)

[4. Data Design 9](#_Toc43596974)

[4.1 Data Dictionary 9](#_Toc43596975)

[5. User Interface 10](#_Toc43596976)

[5.1 Overview of User Interface 10](#_Toc43596977)

[5.2 Screenshots 10](#_Toc43596978)

[5.2.1 Main page 10](#_Toc43596979)

[5.2.2 Register 11](#_Toc43596980)

[5.2.3 Dashboard 13](#_Toc43596981)

[5.2.4 Admin Portal 15](#_Toc43596982)

[5.2.5 Dashboard after approval 17](#_Toc43596983)

[6. Further enhancement 17](#_Toc43596984)

[7. Thoughts 17](#_Toc43596985)

# Table of Figures

[Figure 1 : MERN architecture Diagram 4](#_Toc43596986)

[Figure 2 - Flow diagram of application form filing 6](#_Toc43596987)

[Figure 3 Flow Diagram of Dashboard 7](#_Toc43596988)

[Figure 4 - Flow Diagram of Admin Portal 8](#_Toc43596989)

[Figure 5 Data Dictionary 9](#_Toc43596990)

[Figure 6 Main Page of XYZ portal 10](#_Toc43596991)

[Figure 7 Form Filings with error 11](#_Toc43596992)

[Figure 8 Form Filings with no error 11](#_Toc43596993)

[Figure 9 Form Filing summary 12](#_Toc43596994)

[Figure 10 Form Filing submission successful 12](#_Toc43596995)

[Figure 11 Dashboard checking bookings 13](#_Toc43596996)

[Figure 12 Display all bookings by applicant 13](#_Toc43596997)

[Figure 13 Display individual booking 14](#_Toc43596998)

[Figure 14 When no booking is made 14](#_Toc43596999)

[Figure 15 Admin can view all bookings 15](#_Toc43597000)

[Figure 16 Admin view individual details 16](#_Toc43597001)

[Figure 17 Upon approval, the status change 16](#_Toc43597002)

[Figure 18 Applicant see the updated status at Dashboard 17](#_Toc43597003)

# Introduction

## Purpose

This software design document describes the architecture and system design of an assignment on a web portal created for Company XYZ Pte Ltd.

## Scope

Company XYZ Pte Ltd is a travel agency that helps affiliated companies to make travel arrangements for official purposes.

XYZ-portal is an online portal created to let affiliated companies to put up travel application for their employees and for them to view the status of the application easily. Admin staffs from XYZ are able to use the portal to approve or reject the applications.

## Overview

This document will cover the System architecture, functionalities and UI component of XYZ-portal.

# System Architecture

## Architectural Design

XYZ-portal is made using the MERN stack (MongoDB, Express, React, Node.js). Traditionally, it should be separated into two projects. XYZ-portal is using one project for simplicity with the help of proxy. Everything is hosted on the localhost machine.

The user will be mainly interacting with the applications. React hook is chosen as the front end and node express as the backend. Submission of forms, updating of status will be done through a web service call to the express server. The express server will handle the CRUD of the data.

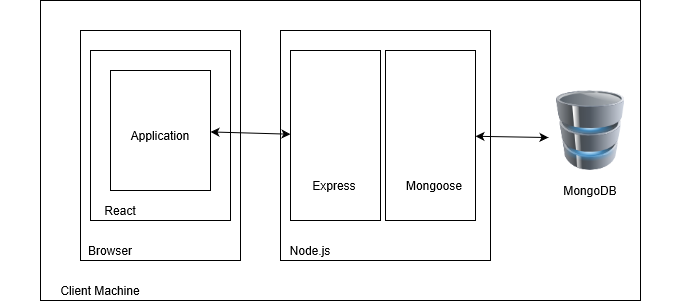


Figure 1 : MERN architecture Diagram

# Functional Requirements

XYZ portal is targeted for two groups of users – applicants from affiliated companies and staffs from XYZ Pte Ltd. Applicants can put up travel applications for employees going overseas for official purpose. After submitting their applications, they can view the costing, as well as the status of the applications they made through a dashboard. Staffs from XYZ Pte Ltd are able to access all applications made by users to approve or reject their applications.

## Application – Form filing

Applicants are able to apply for travel applications via a registration form. In the form, several details are required from them. Company Details, such as name of company, address of company, UEN of company. Applicant Details, such as their name, contact and email address. The details of the employee(s) that the applicant is apply for, such as employee name, employee nric, employee passport number, the country of origin, the country of destination and the travel dates. The applicant is able to register for many employees at the same time using the same form simply by clicking add or minus.

Each field have their own validation check. Most of the fields are mandatory, if the applicant did not fill in, the input field will be highlighted red and error message will appear. Some special checks are done for the company UEN, applicant contact number and applicant email using Regex.

When the applicant clicks continue, and passes all validation checks, he will be brought to a summary page where all the details he keyed in previously will be displayed. The applicant can check on the correctness of his entry before submitting. If any amendment is required, the applicant can go back to the registration form to edit.

After submitting, the data will be sent to the express node server which then will store the data into the database. The costing is only tabulated using a simple random function between the value 1000 and 5000.

A flow diagram describing the process is in Figure-2.

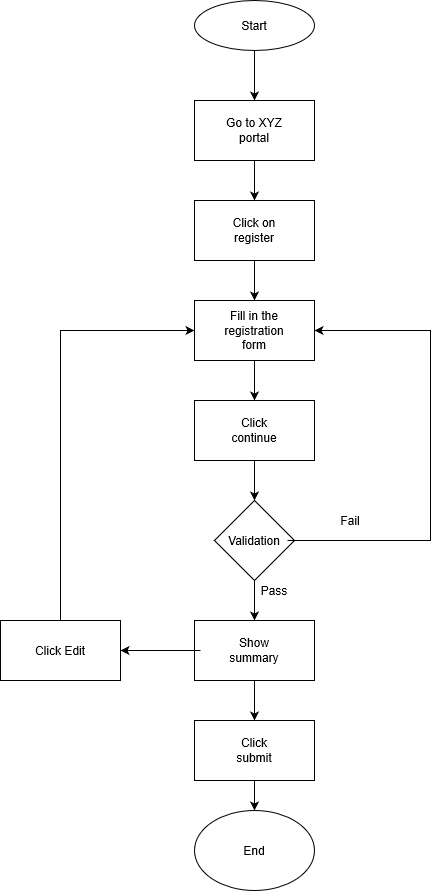


Figure 2 - Flow diagram of application form filing

## Dashboard

Applicants are able to see the past applications through their name and contact number. The name and contact number will be used to query the database for bookings. Once bookings are found, the list of past applications that they submitted will be shown. The latest applications made are shown first.

Applicants will first see a short summary of all their previous application, which include only the booking Id, application status, total costing and the transaction date time. Applicant can click on “View Details” to view more details regarding the application. The individual costing per employee is shown too.

A flow diagram describing the process is in Figure-3.

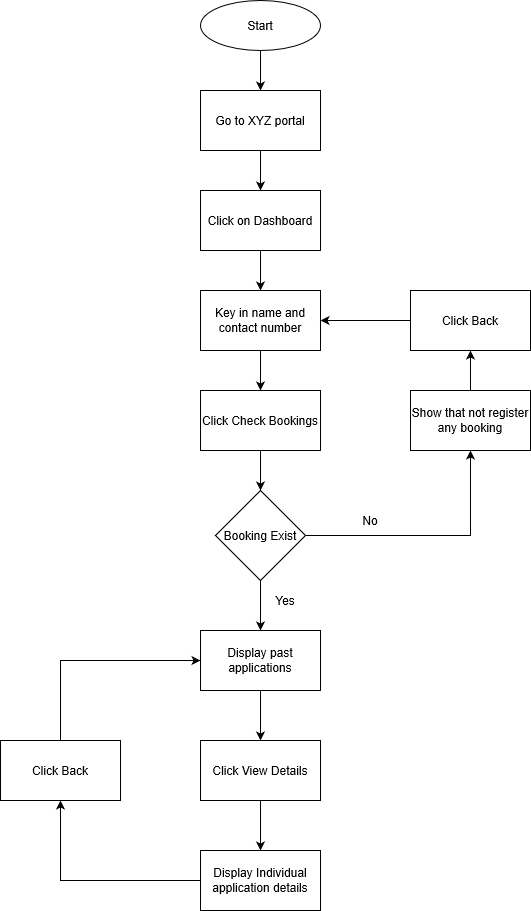


Figure 3 Flow Diagram of Dashboard

## Admin Portal

This admin portal serves as a purpose for staff members to use to approve or reject an application. After accessing the admin portal, all bookings will be displayed. Staff can click on “view details” to view the application details and to approve or reject the application. Once the staff click on either approve or reject, the database will be updated with the value.

A flow diagram describing the process is in Figure-4.

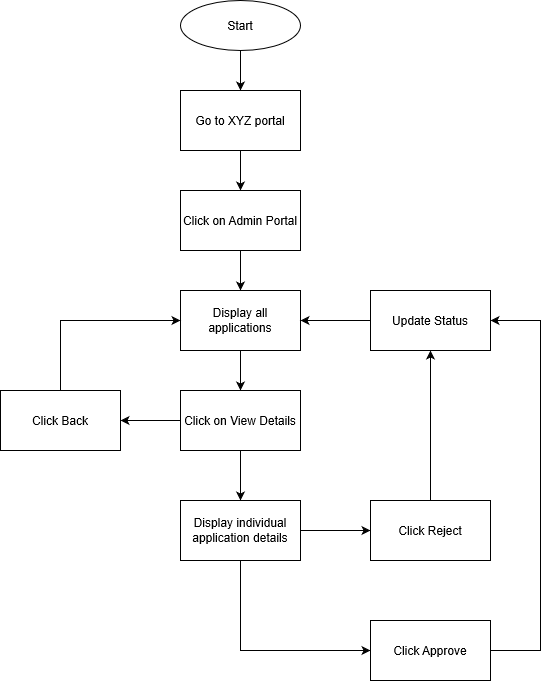


Figure 4 - Flow Diagram of Admin Portal

# Data Design

MongoDB is chosen as the database of XYZ portal for its simple and fast deployment. NoSQL can store different structure of data format. All the data for this XYZ portal is stored under a single schema.

## Data Dictionary

The following describe the data structure being used to store the data.



Figure 5 Data Dictionary

# User Interface

## Overview of User Interface

XYZ portal aims to make its application process and dashboard viewing easy to access. A light blue and white theme is used. As it is a travel agency, light blue and white represents the sky and clouds.

## Screenshots

The followings are the screenshot of the application

### Main page

This is the main page of the XYZ portal.

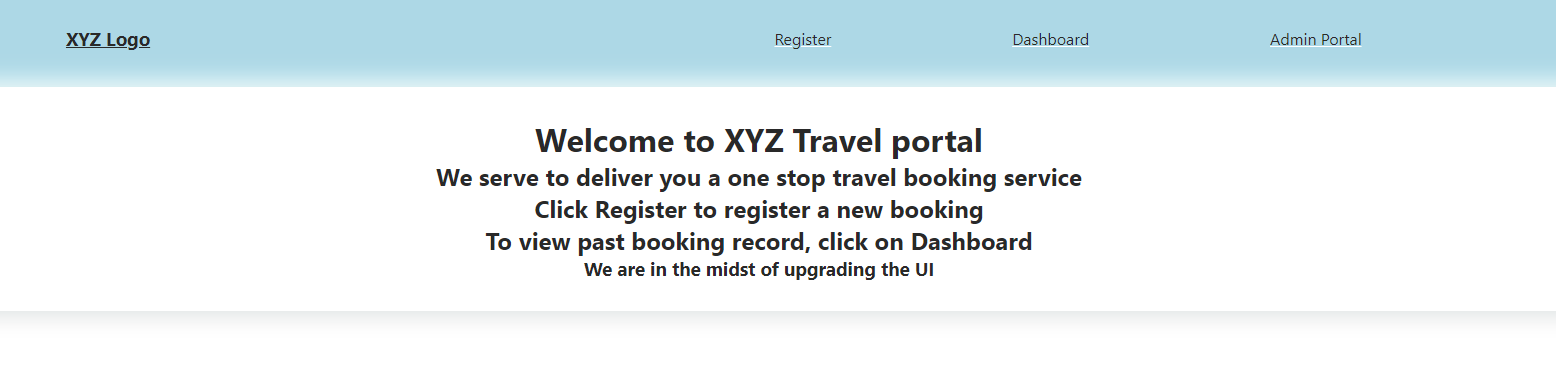


Figure 6 Main Page of XYZ portal

### Register

Applicant can go click on register. If the applicant did not fill in any data and click continue, a validation will be done. Those fields which are incorrect will be highlighted in red, along with an error message.

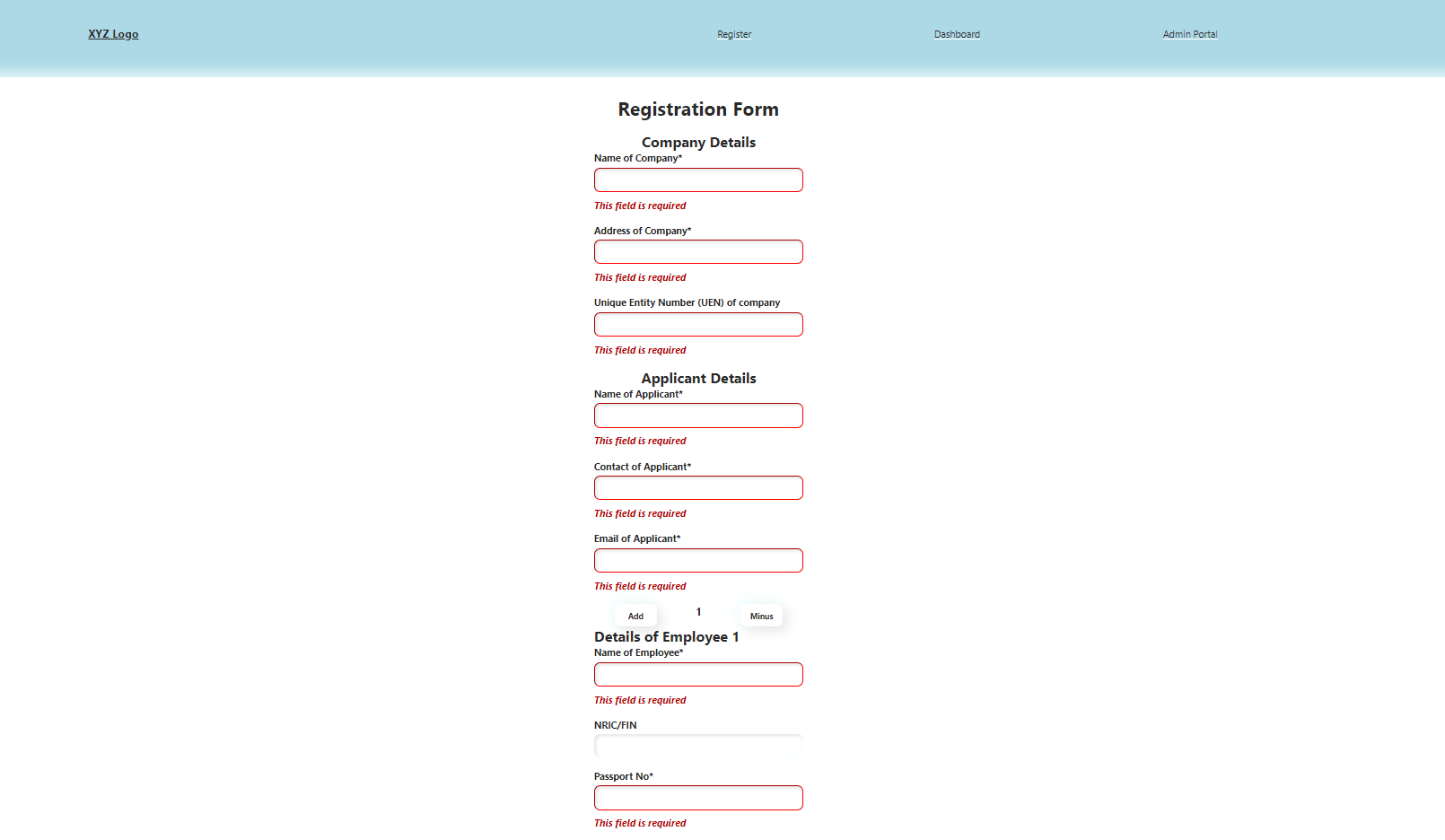


Figure 7 Form Filings with error

The applicant can click on the Add or Minus button to fill in more employee if needed.

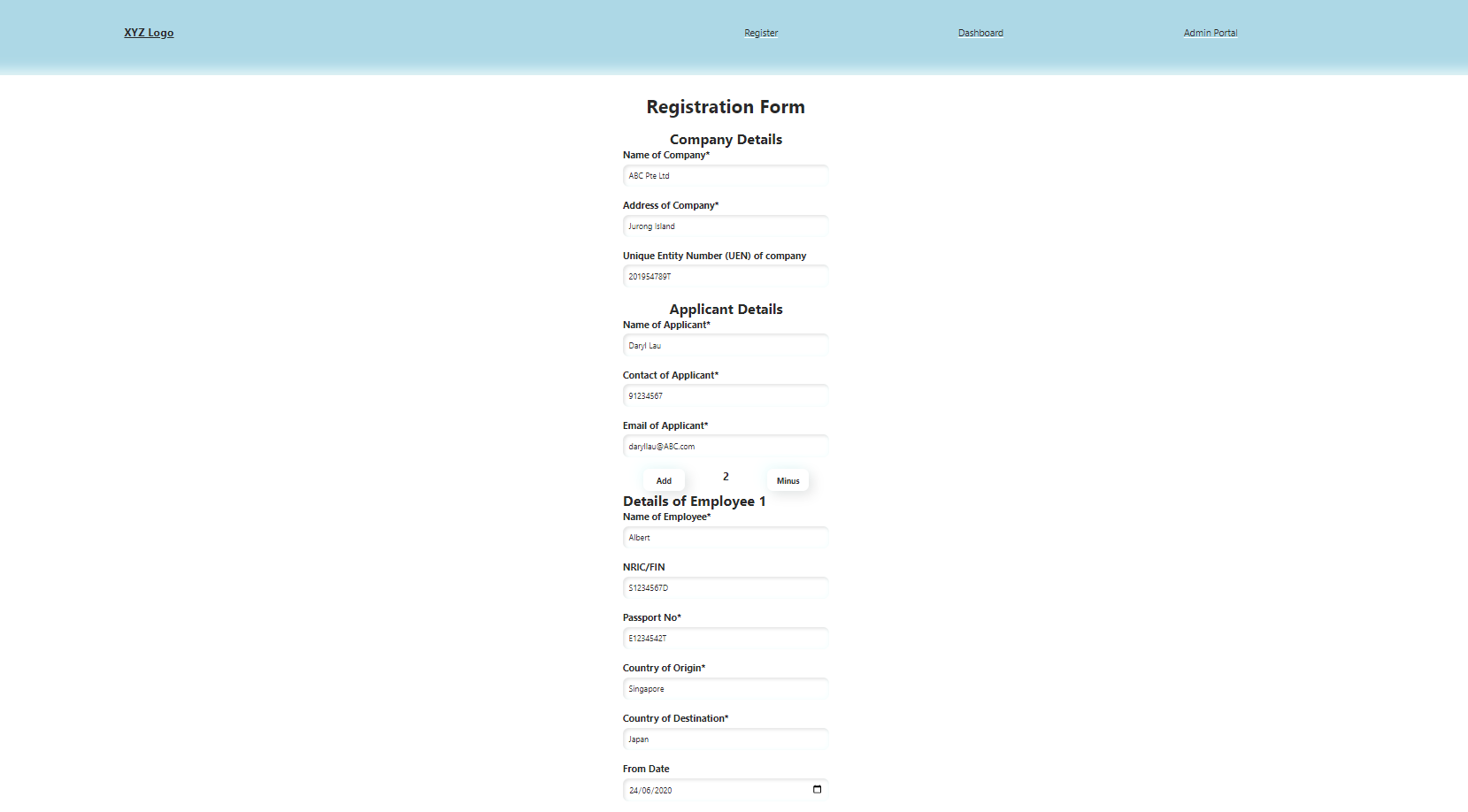


Figure 8 Form Filings with no error

Successful validation will lead to the summary page.

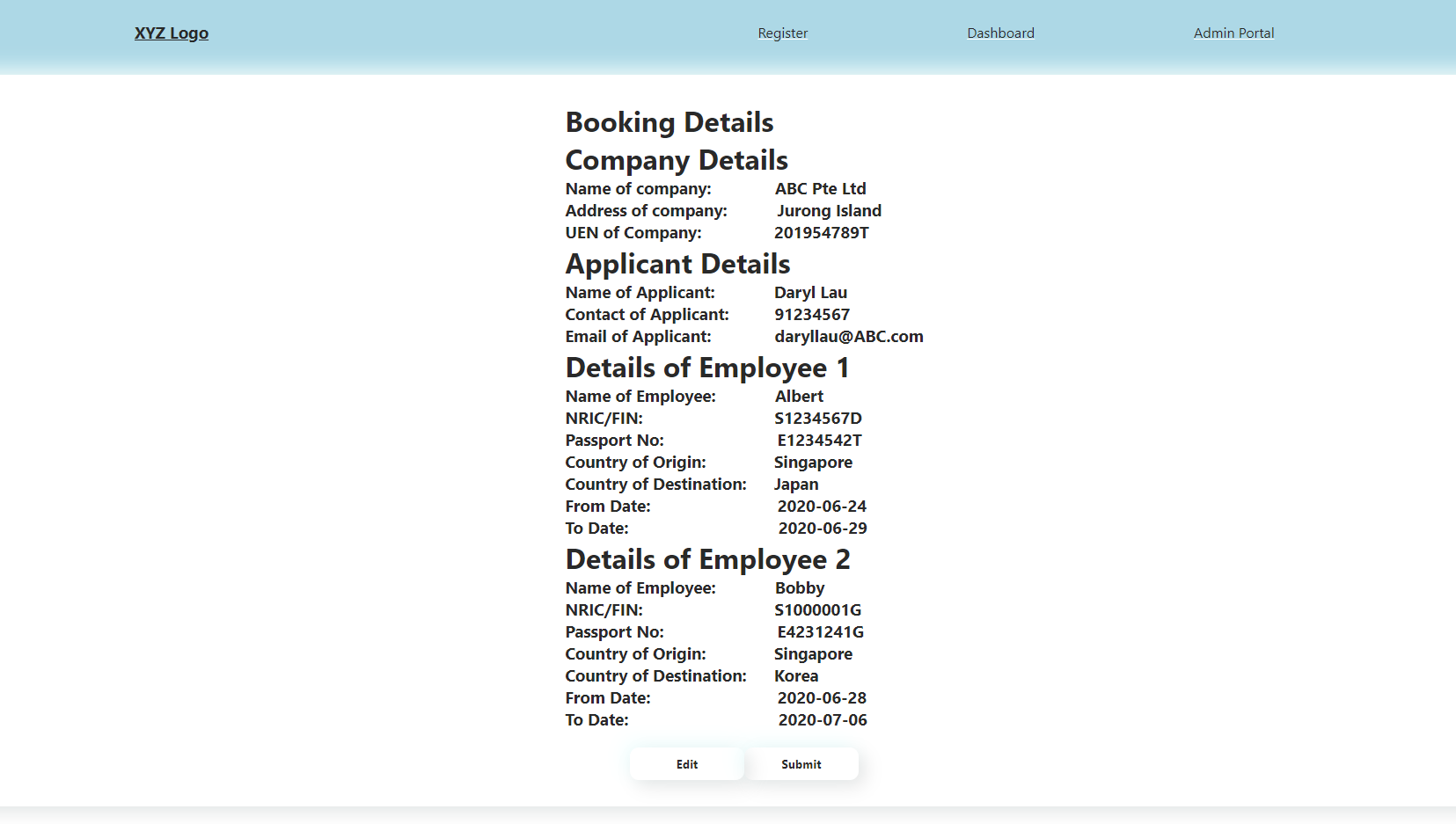


Figure 9 Form Filing summary

After clicking on submit, on successful update, a success message will be shown

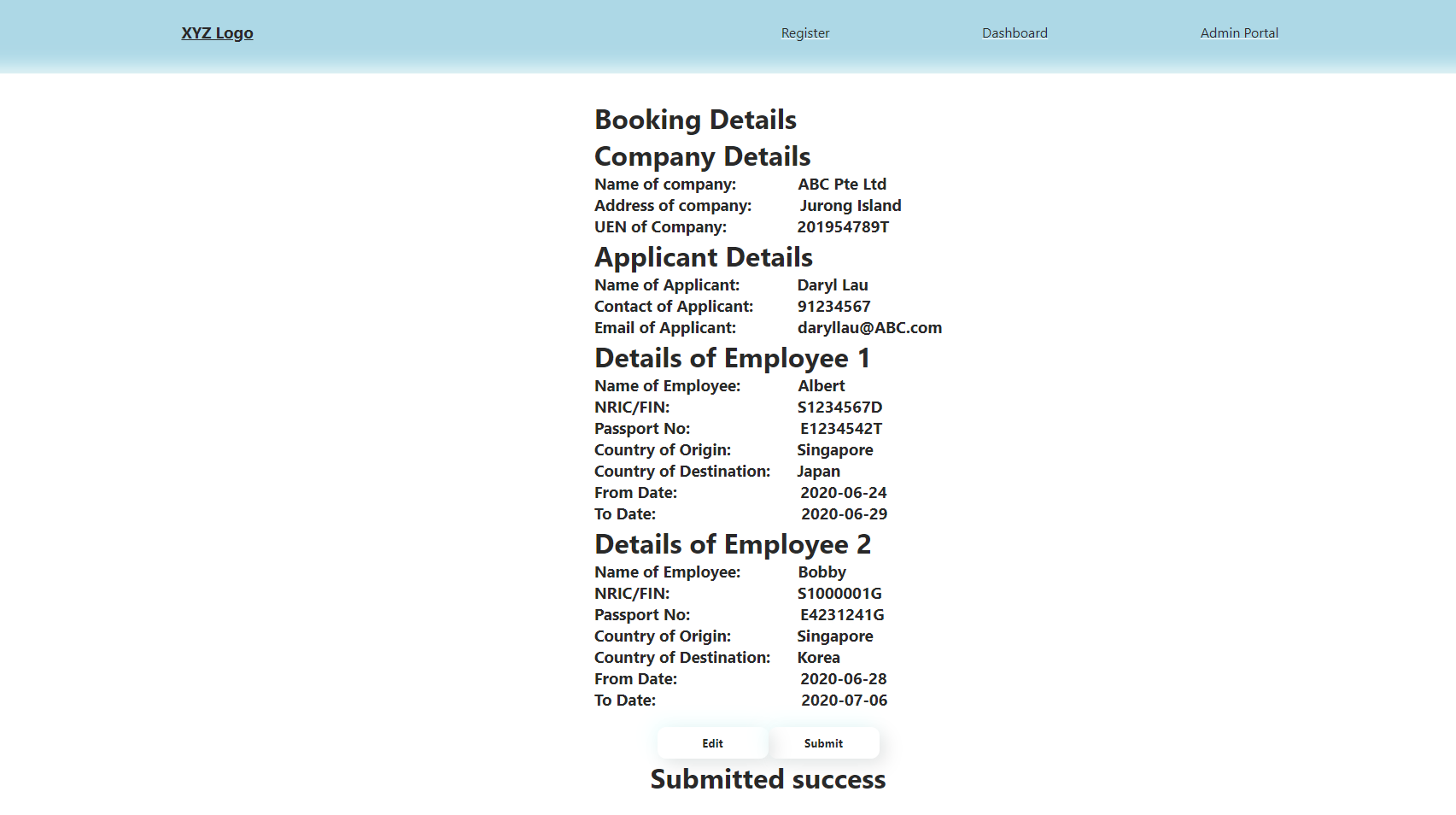


Figure 10 Form Filing submission successful

### Dashboard

The applicant needs to fill in his name and contact number to access his previous bookings.

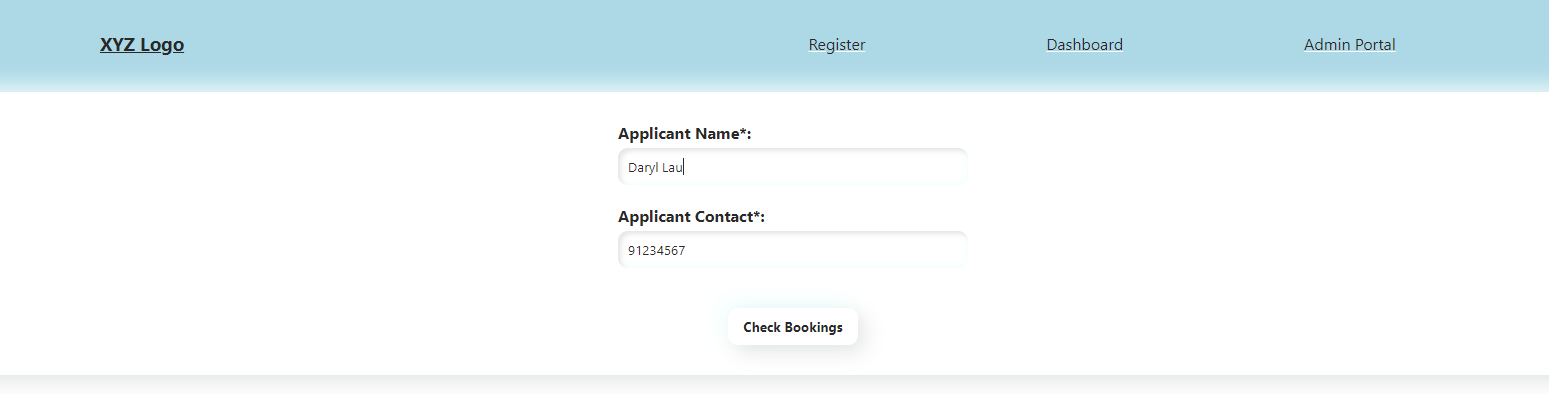


Figure 11 Dashboard checking bookings

Upon successful entry, he can see all his previous bookings as well as the costing and the status.

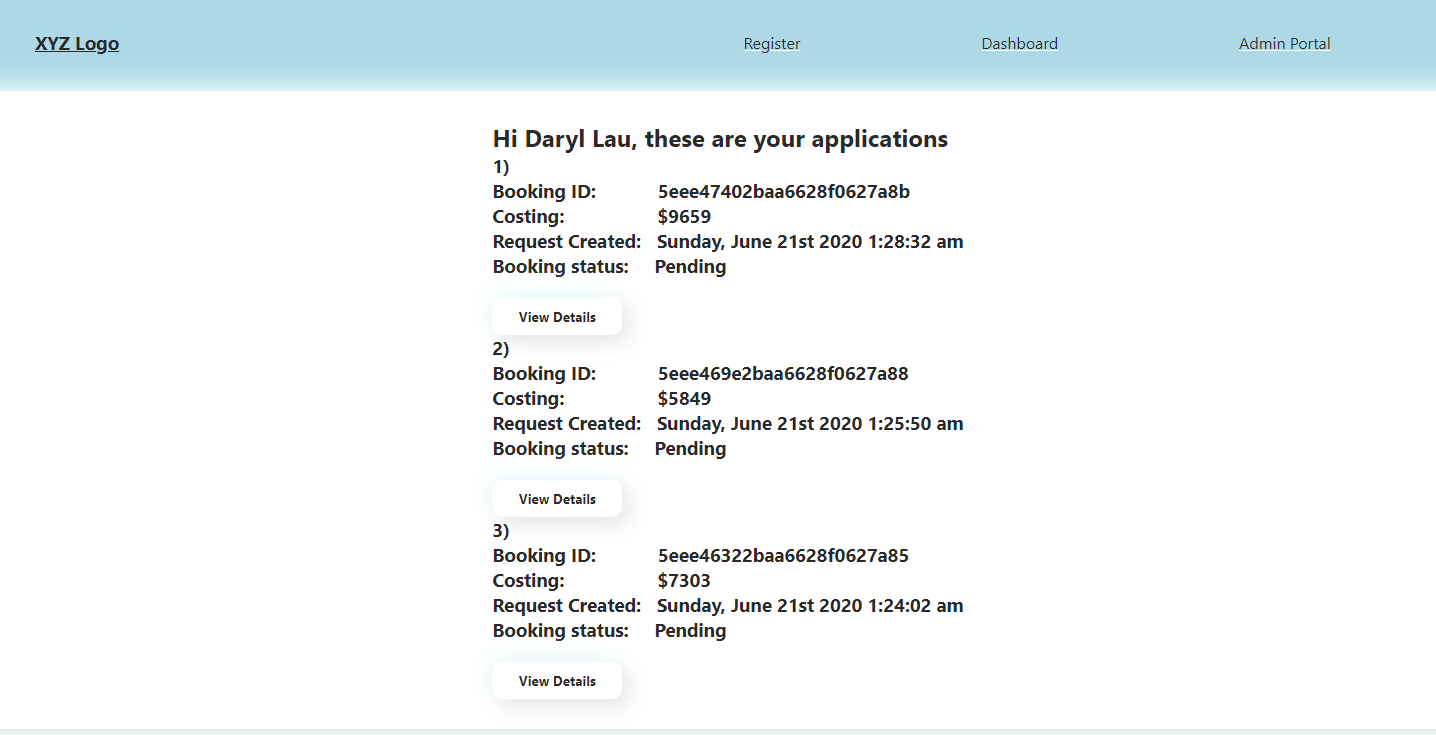


Figure 12 Display all bookings by applicant

The applicant can click on View details to view the individual booking details.

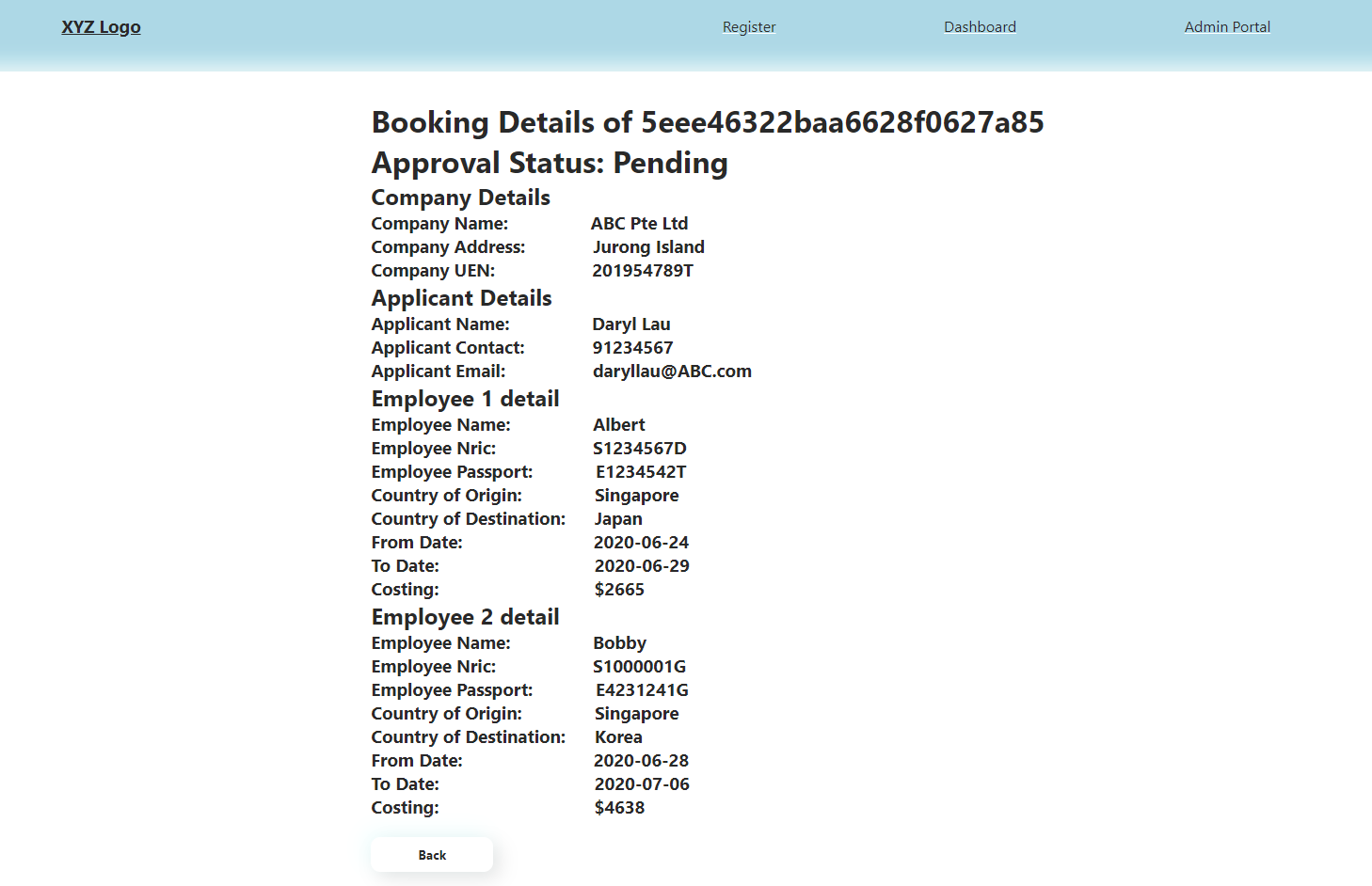


Figure 13 Display individual booking

In the event that the applicant key his name and contact number wrongly, the system cannot find any booking. This error will be prompted.

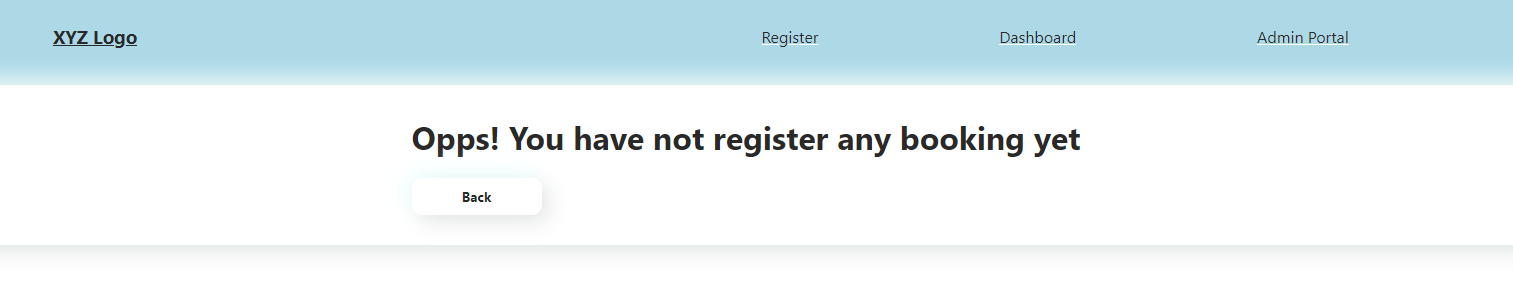


Figure 14 When no booking is made

### Admin Portal

This is what the staff will see when click on the admin portal.

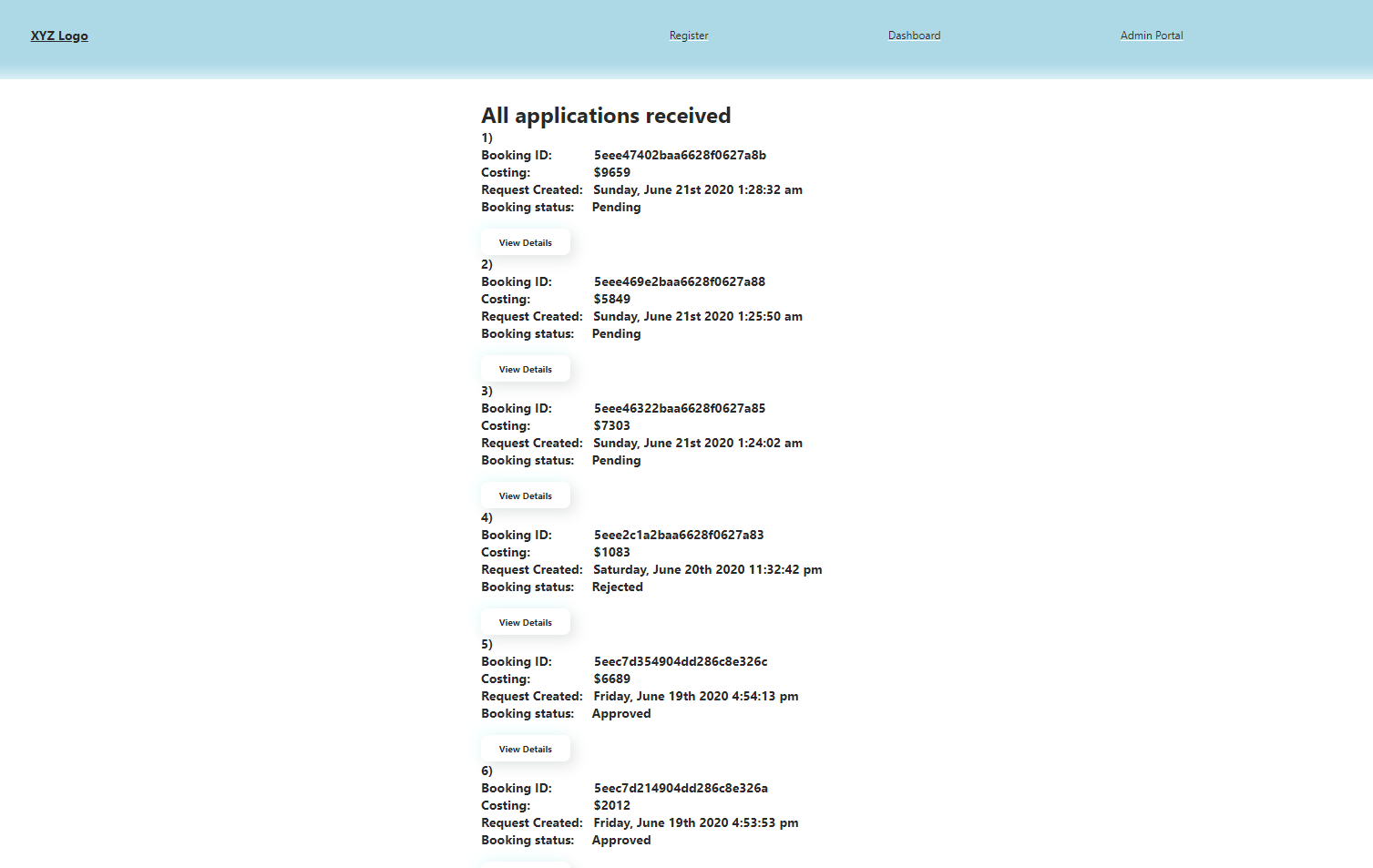


Figure 15 Admin can view all bookings

The staff can click on “view details” to view the details and decide whether to approve or reject the booking

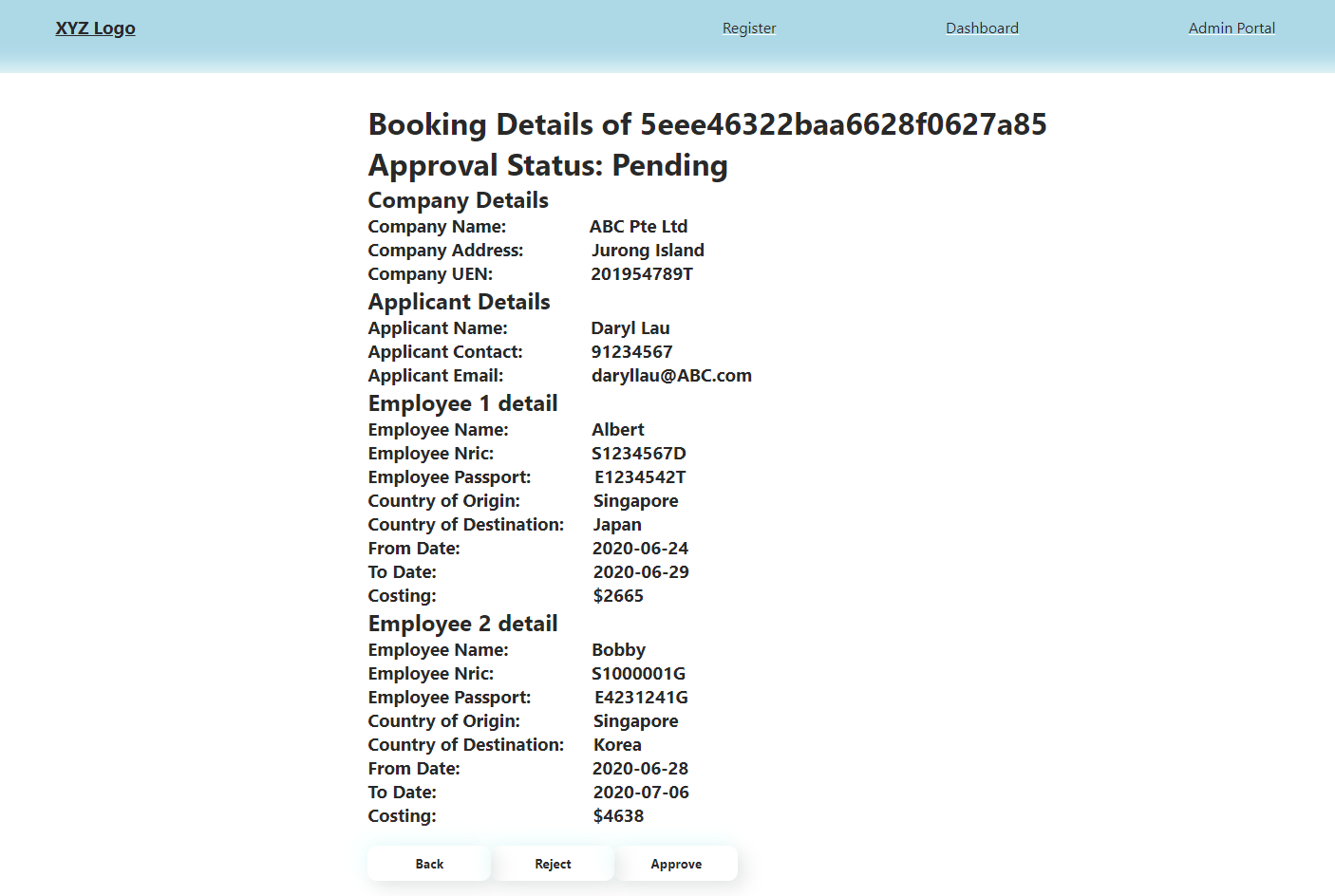


Figure 16 Admin view individual details

The status will be updated if the staff click on Reject or Approve.

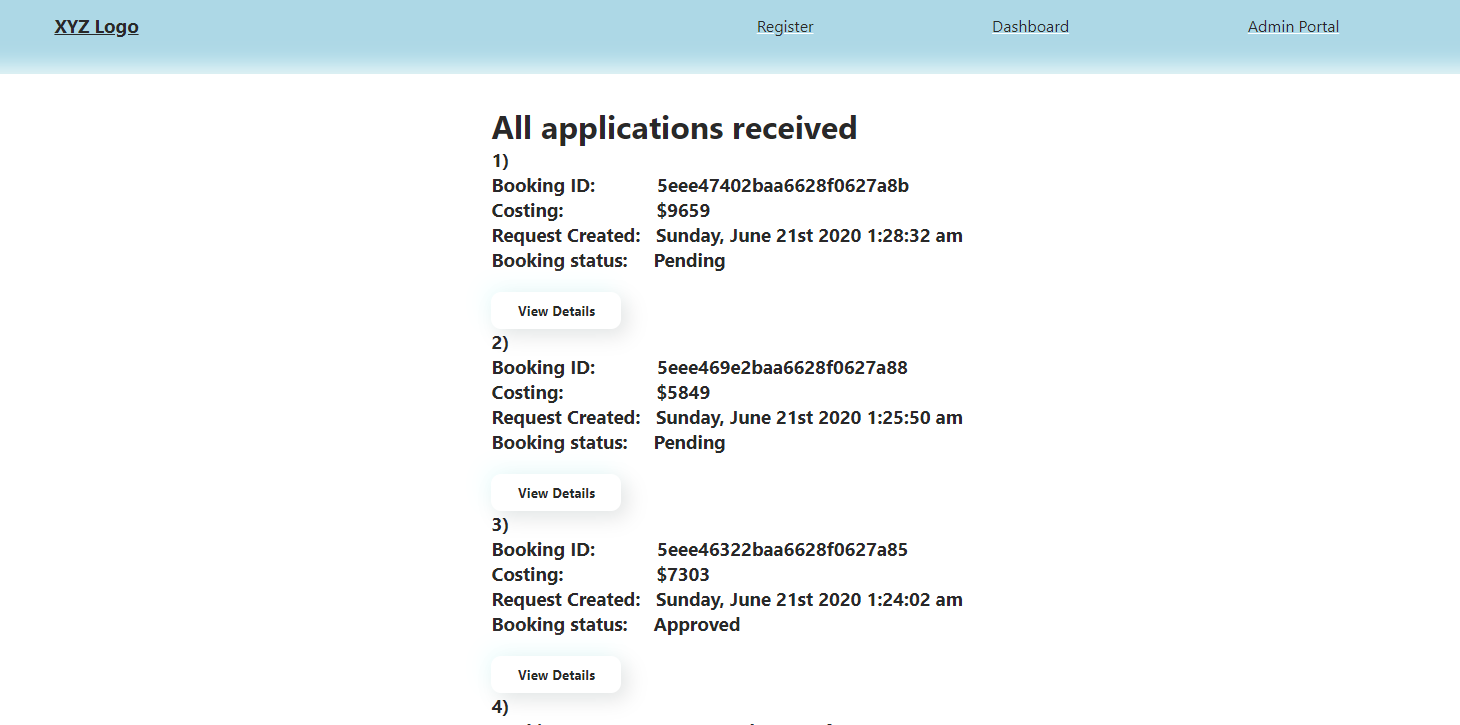


Figure 17 Upon approval, the status change

### Dashboard after approval

Applicant can then go to the dashboard and see the status of the booking.

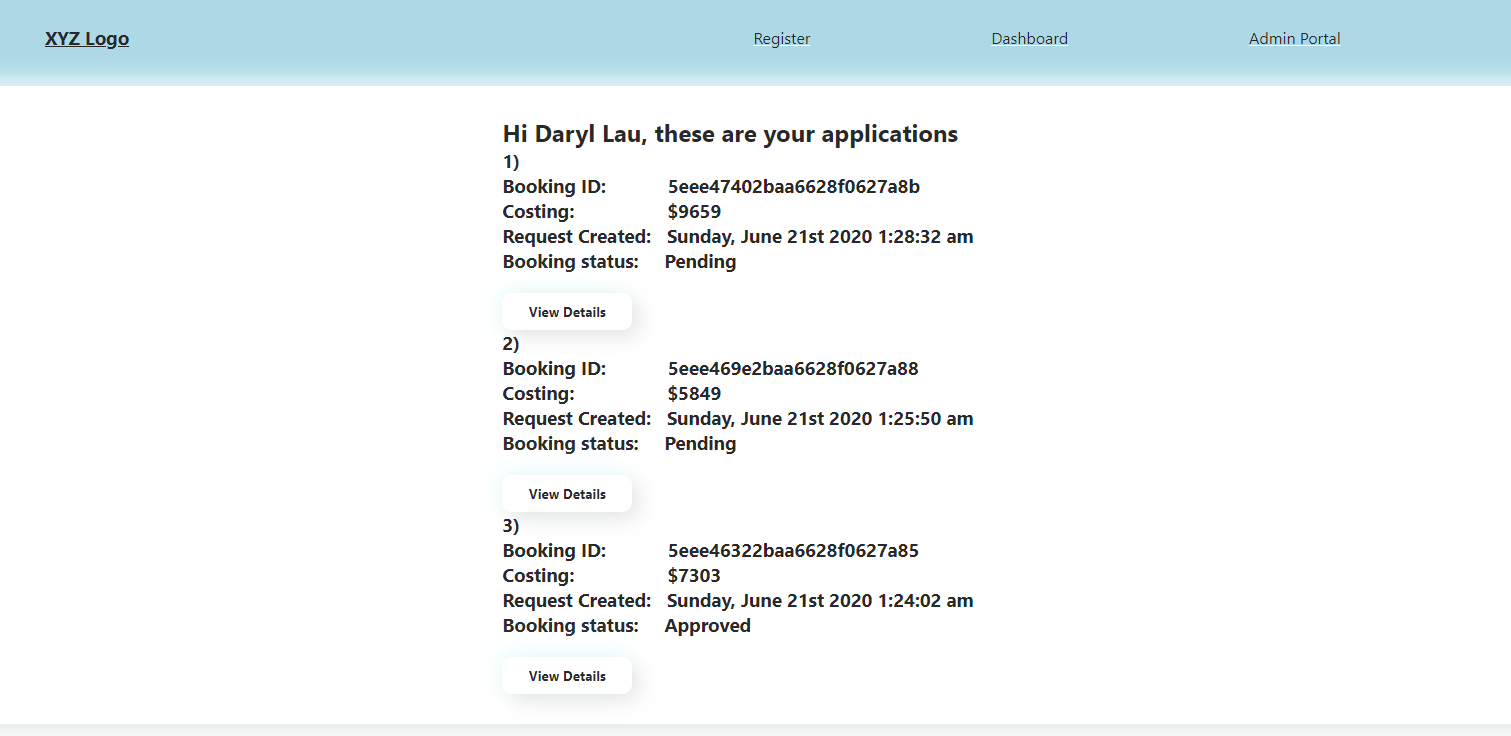


Figure 18 Applicant see the updated status at Dashboard

# Further enhancement

The most needed enhancement is the UI portion.

Next, the thing to change is the employee country of origin and destination country. Make it into autocomplete, or drop down based on actual country name. After that using the country name, hook it up to skyscanner api or some distance API to calculate the costing.

At the dashboard, the applicants can still go and edit his entry if the entry is still pending. When the staff update the status, the new status will be rendered and show on dashboard.

The admin portal be a push notification when need application is submitted. When the admin approve or reject to have a remarks textarea for them to fill in.

# Thoughts

This is my first full stack web development project. These two weeks, maybe lesser due to working hours, spent on researching on React, React hooks, React routers, Node Express and MongoDB is quite fruitful and good learning opportunity. It is my first time using all of these tech stack. There are still many things that I do not have the time to research on, like the designing of UI using React and CSS, unit testing using react.