DEAKIN UNIVERSITY

SIT313

iService Web Application Login Page

Azadeh Ghari Neiat

SIT313 - Developing Client-Server Web Applications

iService Web application (Login Page)

Overview

This task intends to provide you with experience in Node.js, Express, MongoDB and Heroku. You are given the requirements of a Login page for iService web application and deploy your app to Heroku. Your task is to build this component.

You will find “Demo Videos” of Week 2-5 Practical on the unit site to be particularly useful as a reference for this task. Please also keep an eye on your email and any announcements that may be made on Cloud Deakin.

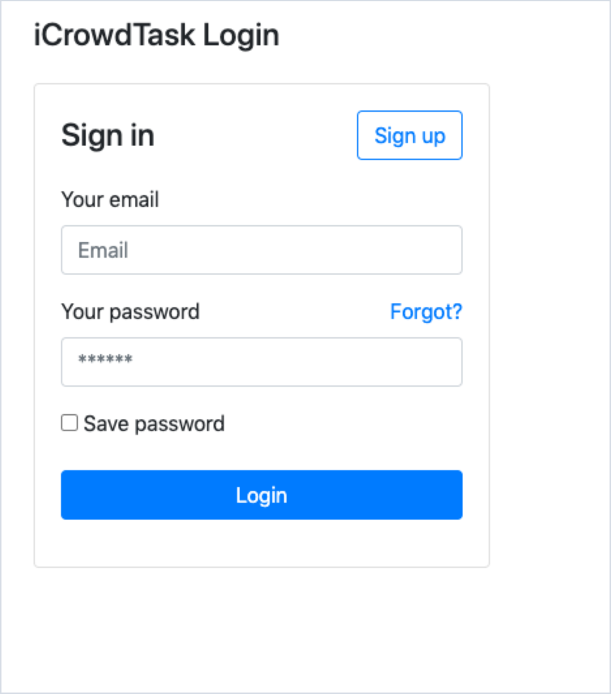
Submission Details

You must submit your app link which needs to be deployed to Heroku and GitHub/Bitbucket link. Please use gitignore to ignore **node_modules** directory. A link to the demo video of your webpage and welcome email are loading and running must be submitted. Please make sure that I and Dinesh have access to the GitHub/Bitbucket. Please put all links in one PDF file and submit it using the task submission page to OnTrack. You must also submit only server.js file to the OnTrack. This is an **individual** assignment, and you should submit **by 8pm AEST, Friday, 3 September 2021, (Week 7).**

Specifications

After registration, a customer or expert needs to log into their iService Account. The Login page allows a customer or expert to log in to their existing account or redirect to the sign-up page (*/custsignup.html*) to create a new account that you have developed in Task 4.1P. There is an HTML template from bootsnipp which is available on OnTrack as a starting point for this task (see Figure 1). You could also use your own nice login page. Feel free to use the Internet for inspiration.

Figure 1. iService Login Page



iCrowdTask Login

Sign in

Your email

Your password

☐ Save password

- As specified in Task 4.1P, the application maintains a MongoDB database to record all information of customers, tasks and experts (*iServiceDB*). In this task, all passwords need to be hashed before storing in the *iServiceDB* database. You could use bcrypt npm library or other packages to hash passwords. According to Wikipedia, "***bcrypt*** *is a password-hashing function designed by Niels Provos and David Mazières, based on the Blowfish cipher*".
- A customer needs to provide their email and password to get into the application. The inputted email and password are checked against the *iServiceDB.* If the customer's login information exists in the database, it redirects the customer to the tasks' page *(/custtask.html*) which is an empty page for now; otherwise, login failure message will be displayed for invalid email or password.
- A login button or link should be placed on the customer registration page in Task 4.1P. Therefore, once a customer creates a new account, they could be re-directed to the *custlogin.html* page. You could also design in a way that when a new customer registers successfully, they will be automatically re-directed to the *custlogin.html* page.
- You need to deploy your app to Heroku and submit the link to your app.