AJAY KUMAR GARG ENGINEERING COLLEGE GHAZIABAD

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

27th KM Milestone, Delhi --Meerut Expressway, Ghaziabad, Uttar Pradesh 201015

A

INTERNSHIP /Mini Project PROJECT REPORT

ON

RESUME BUILDER

SUBMITTED BY

NAME :- SHADAB

YEAR :- 3

SEMESTER 5

SECTION CSE-3

BRANCH CSE

ROLL Number: 2100270100143

DECLARATION

I hereby declare that this submission is my own work and that, to the best of Computer Science and Engineering, it contains no material previously published or written by another person nor material which to a substantial extent has been accepted for the award of any other degree ordiploma of the university or other institute of higher learning, except where due acknowledgment has been made in the text.

Signature: *Shadab,*
Name: *Shadab,*
Roll No: *2100270100143,*
Date *:11/12/2023*

COMPANY CERTIFICATE


udemy

Certificate no: UC-68fb579d-8337-4b40-8f45-f7af3095b6a0
Certificate url: ude.my/UC-68fb579d-8337-4b40-8f45-f7af3095b6a0
Reference Number: 0004

CERTIFICATE OF COMPLETION

The Web Developer Bootcamp 2023

Instructors Colt Steele

SHADAB

Date Nov. 15, 2023
Length 74 total hours

Type

CERTIFICATE

This is to certify that the Mini Project/Internship Assessment Report

entitled “Far Away” which was submitted by Shadab in partial fulfillment of the requirements for the award of the degree B. Tech. in the Department of Computer Science and Engineering of Ajay Kumar Garg Engineering College Ghaziabad, affiliated with Dr. APJ Abdul Kalam Technical University, Uttar Pradesh, Lucknow, is a record of the candidate’s own work carried out by him/her under my supervision. The matter embodied in this report is original and has not been submitted for the award of any other degree.

Supervisor Signature
Supervisor Name:
Date: 15/11/2023

ACKNOWLEDGMENT

I would like to express my sincere appreciation to the individuals and organizations who have contributed to the successful realization of the Far Away Application. Their support, guidance, and encouragement have been instrumental in shaping the project.

I extend my deepest gratitude to Asst. Professor for their unwavering support and guidance throughout the project. Their expertise and insightful feedback significantly influenced the project's development, contributing to its overall success.

My heartfelt thanks go to my friends and family for their steadfast support and understanding during the course of this project. Their encouragement and belief in my capabilities have been a constant source of motivation.

I express my gratitude to Ajay Kumar Garg Engineering College for providing a conducive environment and necessary resources for the successful completion of this project. The facilities and support from various departments were invaluable to the development process.

A special acknowledgment is extended to the creators and maintainers of React. Their contribution provided a robust foundation for specific components of the project, enhancing its functionality and efficiency

ABSTRACT

A resume is a document used by individuals to present their background and skill sets. A resume, also spelled resume or resume also called curriculum vitae (CV. A document that has a brief summary or listing of relevant education and experience. The resume or CV is typically the first item that a potential employer encounters regarding the job seeker and is mostly used for screening an applicant, which is often followed by an interview, while seeking employment in the job search process. A well-designed resume. The Resume Builder will help users build their personal advertisement through the Resume Builder system, which a resume builder with a job placement system. Many large employers use electronic resume processing systems to handle a large number of resumes. Job portal advertisements may direct applicants to email their resumes to the company or visit the company's website and submit a resume in electronic format. Searching for jobs online through popular websites is beneficial, as they have served for so many years as a prominent search tool for job seekers and employers alike. In spite of their valuable utility in linking employers with potential employees, the searching process and technology used by job searching websites have not kept pace with the rapid changes in computing capability and machine intelligence. The information and data retrieval techniques used by these websites primarily depend on manually entered search queries, with some advanced similarity metrics for ranking search results.

Introduction:

In recent years, there has been a continuing trend among youths to pursue higher education in their zeal to become highly qualified and skilled. The new technologies, especially the internet, have made a huge impact on knowledge management and information dissemination in education. In many organizations, including universities, the web portal and knowledge management system are among most popular topics.

Universities have been at the forefront of website development, which has further led to the development of web portals providing more useful links to information resources. Portals have different applications and services to solve various problems. One of the aims of web portals is to enable access to and sharing over the Internet.

For e.g. In a university, new students in the faculty need access to information resources to select various courses and decide on the different areas and majors available in the faculty. This need can be addressed through a knowledge portal which should contain appropriate data about the requirements of the students and users.

The increased number of jobless youths and graduates has become one of the serious issues existing in both developing and developed countries today. The internet has changed the way of looking for employment through the development of online job portals.

A job portal is a type of web portal that provides an efficient way to search the internet or the web for vacant job positions available. This research will go through various types of web/job portals but will, in exact terms, look at job portals as a knowledge management system based on a standard framework. This project will mainly focus on the data and information on available jobs, as needed by unemployed or job seekers.

The web portals have become more important than ever because of the need to access information and gain knowledge using the internet. The existing portals and websites are deeply studied to develop the conceptual framework for the web portal to be developed in this project. Our proposed system is beneficial to everyone for better services in placement.

PROJECT SCOPE AND FEATURES

The development of the React-based travel Todo-list application encompasses a comprehensive set of features carefully designed to address the specific needs and challenges associated with travel planning. The project scope outlines the boundaries and objectives of the application, while the features represent the core functionalities that users can leverage to enhance their travel organization experience.

Project Scope

1.Todo-List Creation and Management

The primary focus of the application is the creation and management of to-do lists tailored for travel purposes. Users can initiate new lists for each trip, capturing a detailed inventory of items they need to pack.

2.Itemization with Quantities

To facilitate thorough planning, the application allows users to itemize their lists with detailed descriptions and associated quantities. This ensures that users have a comprehensive overview of the items they intend to carry, promoting meticulous preparation.

3.Visual Packing Progress

A distinctive feature of the application is the ability to mark items as packed. This visual representation of packing progress provides users with a quick glance at the completion status of their preparations, minimizing the risk of forgetting essential items.

4. Dynamic Sorting Capabilities

Recognizing the diverse priorities of individual travelers, the application introduces dynamic sorting capabilities. Users can arrange their Todo-lists based on different parameters such as item description, quantity, and the time of addition. This customization empowers users to organize their lists in a way that aligns with their unique preferences.

5. One-Click Clearing

Efficiency is a key consideration in travel preparations. The application

streamlines the process of initiating a new list by incorporating a one-click clear function. This feature enables users to reset their to-do lists with ease, providing a fresh starting point for each travel endeavor.

6. Intuitive User Interface:

A user-friendly interface is paramount to the success of the application. The design prioritizes simplicity without compromising functionality, ensuring that users of varying technical proficiencies can navigate the application seamlessly.

7. Responsive Design

Considering the diverse devices and screen sizes used by individuals, the application is developed with a responsive design. This ensures a consistent and optimal user experience across desktops, laptops, tablets, and smartphones.

8.Robust Data Management

To enhance the reliability of the application, robust data management has been implemented. The application securely stores user data, preserving Todo-lists across sessions and devices.

9. Testing and Quality Assurance

Quality assurance is a critical aspect of the project scope. The application undergoes rigorous testing, including unit testing and user acceptance testing, to identify and rectify any potential issues, ensuring a seamless user experience.

10. Future Enhancement Considerations

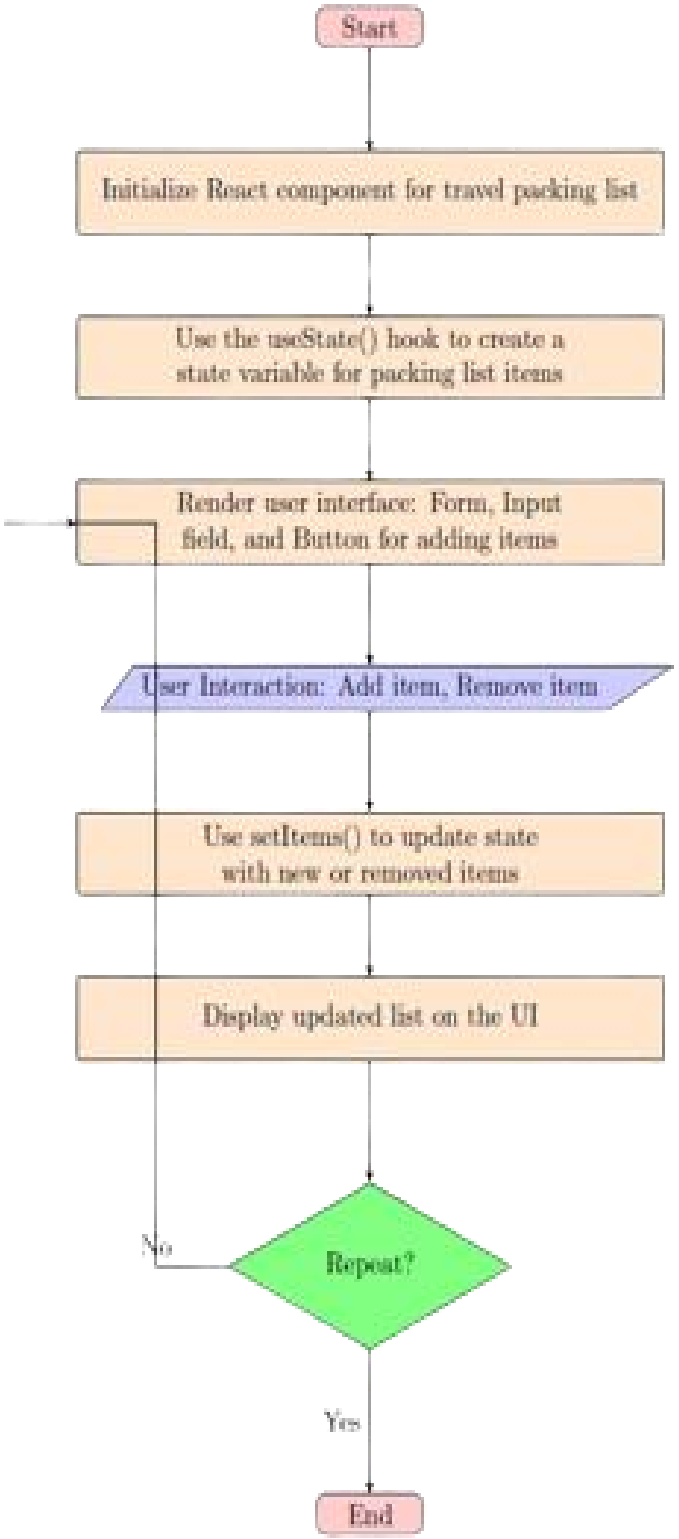
While the primary focus is on the outlined features, the project scope accommodates considerations for future enhancements. The architecture and design are developed with scalability in mind, allowing for the incorporation of additional features and improvements based on user feedback and evolving needs.

Data flow.

- A data flow diagram is a graphical representation that depicts the information flow and the transforms that are applied as data moves from input to output. It can be used to represent software at any level of abstraction. In fact, DFDs may be partitioned into levels. Thatrepresents increasing information flow and functional details. DFDs are defined in levels with every level decreasing the level of abstraction aswell as defining a greater details of the functional organs of the system. A zero-level DFD, also known as a context or fundamental system model, represents all software elements as a single bubble with input and output data entities indicated by incoming and outgoing arrows.

- User interaction:
  - Users interact with the application through the UI, creating, modifying, or accessing their to-do lists.
- Front-end (React):
  - The React frontend captures user input and initiates requests to the backend for data retrieval or storage.
  - The UI components are designed to be reactive, updating in real-time to reflect changes initiated by the user.

Work-Flow of Application



TECHNOLOGIES EMPLOYED

1. React:

React serves as the cornerstone of the front-end architecture. Its component-based structure enables the creation of reusable UI elements, promoting a modular and maintainable codebase. React's virtual DOM (Document Object Model) ensures efficient updates to the UI, optimizing performance.

2. Node.js:

Node.js is utilized to implement the server-side logic of the application. Its event-driven, non-blocking architecture allows for handling multiple concurrent requests efficiently. Node. Node. Node. Node.js aligns seamlessly with JavaScript, offering a unified language for both client and server-side development.

3. *MongoDB*

As a NoSQL database, MongoDB is employed to store and manage data in a flexible, scalable, and JSON-like format. Its schema-less nature accommodates diverse data structures, making it well-suited for handling the dynamic nature of Todo-list data.

4. Express.js:

Express.js is utilized as the web application framework for Node.js. It simplifies the creation of robust APIs and handles routing, middleware, and other essential tasks. Express.js complements Node.js in building scalable and efficient server-side applications.

5. JSON Web Tokens (JWT):

JWT is employed for user authentication. It facilitates secure transmission of information between the front-end and the server by generating tokens that include user credentials. These tokens are validated during each request, ensuring authorized access to user-specific data.

6. Axios:

Axios is used for handling HTTP requests between the front-end and the back-end. Its simplicity and flexibility make it a suitable choice for making asynchronous requests and handling responses.

8. Responsive Design (CSS, Bootstrap).

To ensure a consistent and optimal user experience across various devices, responsive design principles are employed. CSS is used for styling, while Bootstrap, a front-end framework, helps in creating a responsive and visually appealing UI.

IMPLEMENTATION

Implementing the React-based travel to-do list application involves a detailed examination of various aspects, including the development process, coding practices, database integration, user authentication, and testing methodologies. This section will comprehensively explore the implementation of the project, providing insights into the decision-making processes, challenges faced, and the iterative evolution of the application.

Development Process

1. Requirement Analysis:

The journey commenced with a thorough analysis of user requirements and expectations. Understanding the nuances of travel preparations and the specific features users desired informed the initial blueprint for the application. This stage involved close collaboration with potential users to capture diverse perspectives and refine the feature set.

2. Design Phase:

The design phase focused on creating a blueprint for the application's architecture, user interface, and data flow. Wireframes and mockups were crafted to visualize the user experience, ensuring a seamless and intuitive design. Decisions were made regarding the choice of technologies, including React for the front-end, Node.js for the server, and MongoDB for data storage.

3. Front-End Development (React):

The front-end development commenced with the creation of a modular and reusable component structure using React. The application's user interface was designed to be clean, intuitive, and responsive, ensuring optimal user experience across various devices. Key components included forms for adding items, visual displays of to-do lists, and interactive features for marking items as packed.

4. Back-End Development (Node.js):

Node.js was employed for the server-side development, handling requests from the front-end, processing business logic, and communicating with the database. Node. Node. Express.js facilitated the creation of a robust API, defining routes for data retrieval, storage, and user authentication. The back-end ensures the security and integrity of user data by implementing measures to prevent unauthorized access.

5. Database Integration (MongoDB):

MongoDB served as the database for the application, storing user information, Todo-lists, and associated details. The flexible schema of MongoDB accommodates the dynamic nature of Todo-list data. Mongoose, an ODM library, facilitates interactions with the database, defining schemas and models for structured data management.

6. User Authentication (JWT):

User authentication is a critical aspect of the implementation to secure access to user-specific data. JSON Web Tokens (JWT) are employed to generate tokens containing user credentials. These tokens are validated during each request, ensuring that only authorized users can access and modify their Todo-lists.

7. Sorting and Filtering Module:

The sorting and filtering module is implemented to enhance user customization. Users can dynamically sort their Todo-lists based on different parameters, such as item description, quantity, and time of addition. This module seamlessly integrates with the API to process user requests for personalized organization.

8. Responsive Design (CSS, Bootstrap).

The user interface is designed with responsiveness in mind, utilizing CSS for styling and Bootstrap to create a visually appealing and adaptive design. The application's layout adjusts seamlessly to different screen sizes and devices, ensuring a consistent and optimal user experience.

import

Navbar.js

```
import { Box, Flex, HStack, IconButton, useDisclosure, useColorMode,
useColorModeValue, Stack, Button } from '@chakra-ui/react';
import { HamburgerIcon, CloseIcon, MoonIcon, SunIcon } from '@chakra-
ui/icons';
import { Link as ReachLink } from 'react-router-dom';

import logo from './../../Assets/logo.png';



export default function Navbar() {
   const { colorMode, toggleColorMode } = useColorMode();
   const { isOpen, onOpen, onClose } = useDisclosure();

   return (
      <>
         <Box id='navbar' bg={useColorModeValue('gray.100', 'gray.900')}
px={4}>
            <Flex h={16} alignItems={'center'} justifyContent={'space-
between'}>
               <ReachLink to='/'>
                  <Box><img style={{ height: '44px' }} className='logo'
src={logo} alt="logo" /></Box>
               </ReachLink>

               <HStack spacing={8} alignItems={'center'}>
                  <HStack
                     as={'nav'}
                     spacing={4}
                     display={{ base: 'none', md: 'flex' }}>
                     <ReachLink px={2} py={1} rounded={'md'}
_hover={{ textDecoration: 'none', bg: 'gray.200' }} to={'/'} >Home
```

```
</ReachLink>
                    <ReachLink px={2} py={1} rounded={'md'}
_hover={{ textDecoration: 'none', bg: 'gray.200' }} to={'/about'}>
About</ReachLink>
                </HStack>
                <Button onClick={toggleColorMode}>
                    {colorMode === 'light' ? <MoonIcon /> : <SunIcon />}
                </Button>
            </HStack>

            <IconButton
                size={'md'}
                icon={isOpen ? <CloseIcon /> : <HamburgerIcon />}
                aria-label={'Open Menu'}
                display={{ md: 'none' }}
                onClick={isOpen ? onClose : onOpen}
            />

        </Flex>

        {isOpen ? (
            <Box pb={4} display={{ md: 'none' }}>
                <Stack as={'nav'} spacing={4}>
                    <ReachLink px={2} py={1} rounded={'md'}
_hover={{ textDecoration: 'none', bg: 'gray.200' }} to={'/'} >Home
</ReachLink>
                    <ReachLink px={2} py={1} rounded={'md'}
_hover={{ textDecoration: 'none', bg: 'gray.200' }} to={'/about'}>
About</ReachLink>
                </Stack>
            </Box>
        ) : null}
    </Box>

    </>
  );
}
```

Footer.js

```
import { Box, chakra, Container, Stack, Text, Image, useColorModeValue,
VisuallyHidden } from '@chakra-ui/react';
import { FaInstagram, FaSnapchat, FaGithub } from 'react-icons/fa';
import logo from './../../Assets/logo.png';


const SocialButton = ({ children, label, href }) => {
   return (


      <chakra.button
         bg={useColorModeValue('blackAlpha.100', 'whiteAlpha.100')}
         rounded={'full'}
         w={8}
         h={8}
         cursor={'pointer'}
         as={'a'}
         href={href}
         display={'inline-flex'}
         alignItems={'center'}
         justifyContent={'center'}
         transition={'background 0.3s ease'}
         _hover={{
            bg: useColorModeValue('blackAlpha.200', 'whiteAlpha.200'),
         }}>
         <VisuallyHidden>{label}</VisuallyHidden>
         {children}
      </chakra.button>

   );
};

export default function Footer() {
```

```
  return (
    <>
      <Box
        bg={useColorModeValue('gray.50', 'gray.900')}
        color={useColorModeValue('gray.700', 'gray.200')}>
        <Container
          textAlign={'center'}
          as={Stack}
          maxW={'6xl'}
          py={4}
          direction={{ base: 'column', md: 'row' }}
          spacing={4}
          justify={{ base: 'center', md: 'space-between' }}
          align={{ base: 'center', md: 'center' }}>
          <Image style={{ height: '44px' }} src={logo} alt="logo" />
          <Text>© 2023 Resume Builder, All rights reserved</Text>
          <Stack direction={'row'} spacing={6}>
            <SocialButton label={'Github'}
href={'https://github.com/imhardikdesai'}>
              <FaGithub />
            </SocialButton>
            <SocialButton label={'Snapchat'}
href={'https://twitter.com/imhardikdesai'}>
              <FaSnapchat />
            </SocialButton>
            <SocialButton label={'Instagram'}
href={'https://instagram.com/imhardikdesai'}>
              <FaInstagram />
            </SocialButton>
          </Stack>
        </Container>
      </Box>

    </>
  );
```

}}

OVERVIEW

Figure 1a: Adding Items

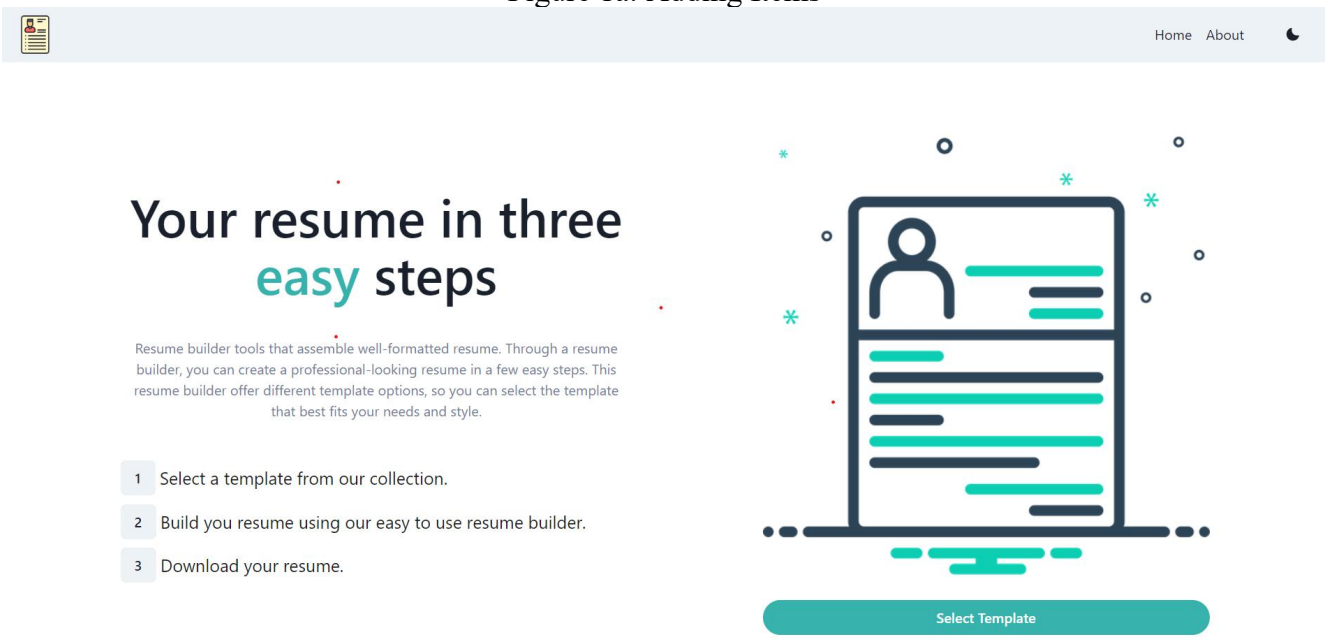


Figure 1b: Cleared Added Items

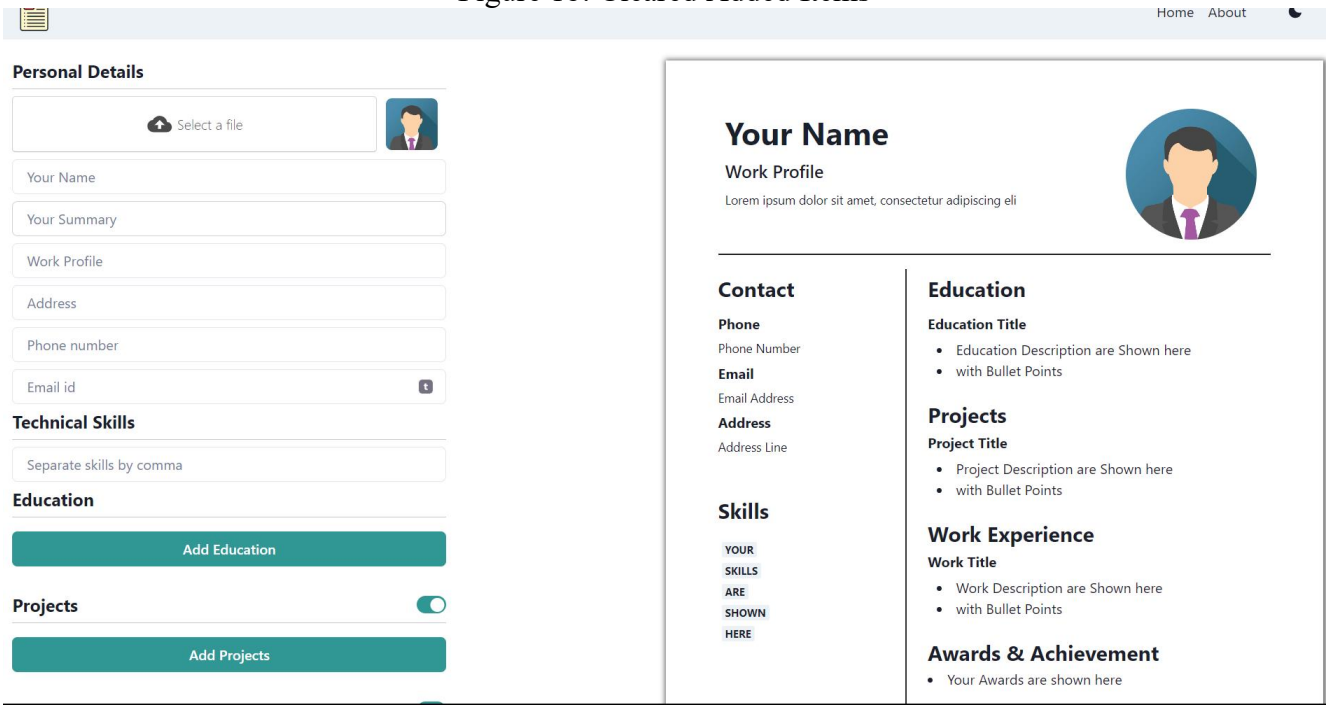
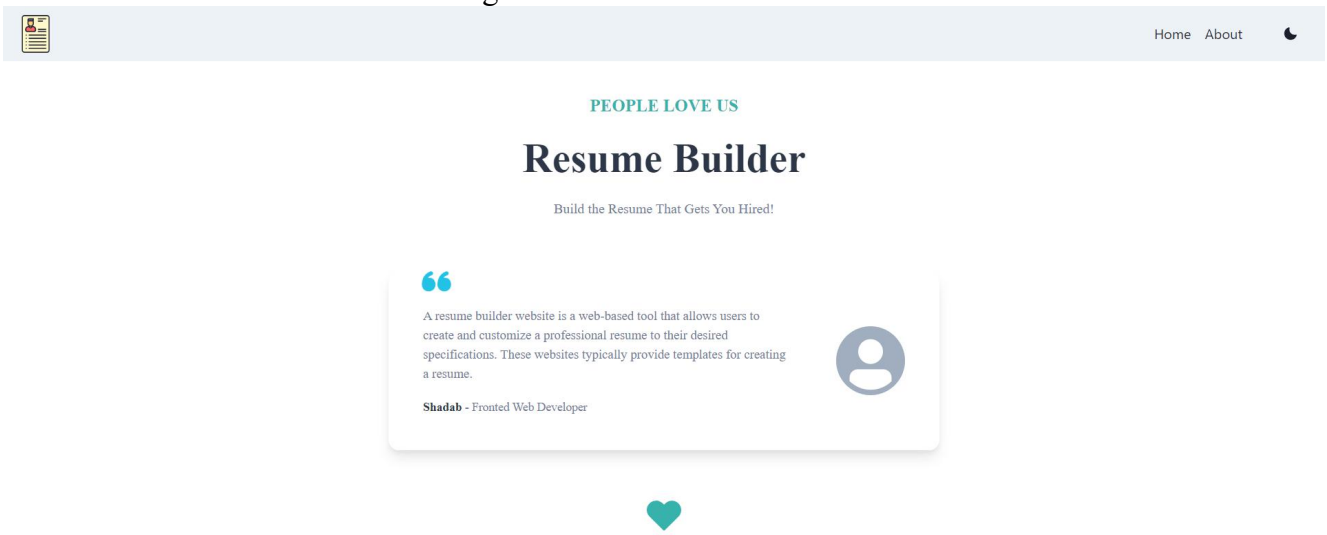


Figure 1c: Marked Packed Items



RESULT AND DISCUSSION

The implementation of the React-based travel to-do list application culminated in a functional and user-centric platform designed to enhance the travel planning experience. This section delves into the results of the implementation, discussing key achievements, user feedback, and the implications of the application in addressing the challenges of travel organization.

Key Achievements

1. User-Friendly Interface:

- The user interface successfully achieved a balance between simplicity and functionality. Users reported a positive experience in navigating the application, adding and managing items, and utilizing sorting features.

2. Real-Time Updates:

- The implementation successfully addressed the challenge of real-time updates. Users could observe instantaneous changes in their to-do lists as items were marked as packed, providing a visual representation of their packing progress.

3. Dynamic Sorting:

- The sorting and filtering module empowered users to personalize their to-do list organization. Users appreciated the flexibility to sort items based on description, quantity, and time of addition, tailoring their lists to meet their specific preferences.

4. Responsive Design:

- The application's responsive design garnered positive feedback, ensuring a consistent and optimal user experience across various devices. Users could seamlessly access and interact with the application from desktops, laptops, tablets, and smartphones.

5. Secure User Authentication:

- The implementation of secure user authentication using JSON Web Tokens (JWT) proved effective in safeguarding user data. The application maintained a robust security posture, preventing unauthorized access and ensuring data integrity.

6. Clear Navigation and Intuitive Controls:

- Users commended the clear navigation and intuitive controls within the application. The design facilitated a straightforward process for creating new to-do lists, adding items, and managing sorting preferences.

7. Iterative Development and User Feedback:

- The iterative development process, guided by continuous user feedback, contributed to the application. Users actively participated in user acceptance testing, providing valuable insights that influenced feature enhancements and improvements.

References:

- Writing Resilient Components (By Dan Abramov from the React team)
- Things I think about when I write React code (GitHub repository,
- A (Mostly) Complete Guide to React Rendering Behavior (by Mark Erikson from the GitHub repository,
- A Visual Guide to React Rendering (a multi-part series, check out the others.
- Inside Fiber: in-depth overview of the new reconciliation algorithm in React
- A Cartoon Intro to Fiber (YouTube video).