**Bicycle Bandits**

<https://youtu.be/Rw1oUMVaRDo>

To Do List

* At least 2 word searching
* Login error if incorrect info is entered
* See more ticket results
* Add suggested ticket time by adding up labor times
* Todo List
  + Add ability to check/uncheck items
* Improve Search
  + Ticket search incorporated into header for employees
  + Search functionality needs to be improved
  + Use reg ex for searches on names / product titles
    - Store names and products without modifying the case
* If you try to add a lobor/product that is already on the ticket then increment quantity by 1 instead of doing nothing (ticketItems.jsx)
* Address Autofill?
* Once the site is pretty refined, start setting up products page to display and search
* Password retrieval functionality
* What other technologies might I want to incorporate?
  + React-cookie
  + React-spinners
  + Sweetalert2
  + Address Autofill Package
  + OAuth2.0

React-icons for any type of icon needed

bicycle-bandits-deployment-access

AKIASNREFPCN3QKNOM44 - Access Ke

7/plc2pEo/mY9t66AwTEKTe+FJKUxMGTMJwzpdcF - Secret access key

Feature Wish List

* HTTPS – secure
* Payments system
* Product API for both customers and employees
* Error Modals
* Success Modals
* Phone Friendly
* Address recommends complete addresses as the user types
* Figure out Redux for Authorization with JWT

Strip Setup

* View cart button redirects to a product display page with all cart information
  + Checkout button redirects to Stripe payments page
* Checkout Success Page / notification (sweet alerts)
* Checkout Cancel Page / notification (sweet alerts)

Goal Product

* Features
  + Users can login/register and logout
  + Customer Account
    - User can view a list of their current and previous tickets
    - User can view the details of any of their tickets
  + Employee Account
    - Tech can view upcoming tickets from home page
      * Can navigate to ticket directly
    - Tech can navigate to a ticket search page and use it to look for specific tickets
    - Tech can create a new ticket
    - Tech can view and edit ticket details
      * Tech can add labor and products to the ticket
* Pages
  + Landing Page
  + Login/Register Page
  + Customer Pages
    - Customer Dashboard Page
    - Customer Ticket Details Page
    - Customer Shopping Page
    - Customer Product Details Page
    - Customer Cart/Checkout Page
    - Customer Ticket Scheduling Page
  + Tech Pages
    - Tech Dashboard Page
    - Tech Ticket Search Page
    - Tech Ticket Details/Edit Page
      * Sub page for customer ticket review
    - Tech New Ticket Page
    - Tech Product/Labor Search Page
* Code/Technology
  + Front end built with:
    - React
    - Navigation with React Router
    - Http Requests with Axios
    - Bcrypt for password hashing / authorization
    - JSON Web Tokens for Authentication
      * Potentially switch to OAuth2
    - Redux for global state management
      * Redux toolkit
    - Styling with Material UI and custom CSS
    - Formik for forms
  + Back end built with:
    - NodeJS
    - Express
    - Cors
    - Sequelize for database queries
    - Bit.io for database

CSS Decisions

* Fonts – from google fonts
  + Header – Rye
  + Main – Montserrat
* Colors
  + Dark - #30362F
  + Mid - #625834
  + Light - #FFFBDB
  + Highlight - #A59132
  + Highlight - #DA7422
* Border Radius
  + 8px

Components Planning

* UI
  + Buttons – Material UI
    - Small
    - Medium
    - Large
  + Container
    - Sub Container
    - Card
  + Input – marterial UI
  + Search Bar
  + Car Line Item
  + Tight Line Item
  + Drop Down Selector
  + Side Bar
  + List Item Container?
* Header
  + Will change depending on screen/user type
* Footer
* Login
  + Landing Page
    - Container
      * Register Button
      * Login Button
  + Login Page
    - Form Container
      * 2 x inputs
      * Button
  + Register Page
    - Form Container
      * 8 x inputs
      * Button
* Customer
  + Dashboard Page
    - Products Preview Container
      * Search Bar
      * Product Cards
    - Ticket Overview Container
      * Title Bar
      * List Items
      * Navigation Buttons
    - Calendar Container
      * Dynamic Calendar
  + Ticket Details Page
    - Primary Container
      * Customer / Ticket Details
        + Ticket Info
        + Customer Info
        + Bike Info
      * Tech Notes
      * Charges Container
        + Title Bar
        + List Items
        + Total/Checkout
  + Shopping Page
    - Primary Container
      * Search Refinement Side-bar
      * Search Bar
      * Product Cards
      * Navigation Buttons
  + Product Details Page
    - Primary Container
      * Product Image
      * Product Highlights
        + Primary details
        + Add to cart button
      * More Details
  + Customer Cart
* Tech
  + Dashboard Page
    - Ticket Overview
      * Title Bar
      * Search Bar
      * Line Items
    - TO DO List
      * Title bar
      * Line items
      * Add new button
    - Calendar
    - Charts Container
      * Daily progress chart
      * Revienue chart
  + Ticket Search Page
    - Primary Container
      * Search refinement side-bar
      * Search Bar
      * Category Selector
      * Search Results
        + Title Bar
        + Line Items
  + New Ticket Page
    - Primary Container
      * Inputs Container
        + Inputs
      * Suggestions Container
        + Suggestion Cards
      * Create Ticket Button
  + Ticket Details Page
    - Ticket Info Container
      * Edit/Delete Buttons
      * Ticket Info
      * Customer details
      * Bike details
    - External Notes
    - Internal Notes
    - Charges Container
      * Header
        + Search Bar
        + Checkout button
      * Title Bar
      * Line Items
      * Footer
  + Prodcut/Labor Search Page
    - Refinement Side-bar
    - Search bar
    - Results Container
      * Title Bar
      * Line Items

End Point Planning

State Planning

Validation & Logic Planning

* Logic for errors
* Validation for inputs

Color Code

* Required Content
* Should Include for best product
* Add if things go well
* Last features to be added
* Technology Being Used

Functionality Planning

* Not Logged In
  + Header
    - Site name/logo
    - Nav Bar
      * Login/Create Account
  + Landing Page
    - Focus on logging in or creating and account
    - Take a tour button
      * Allows user to bypass login/registration and view app as a guest
    - Secondary entertainment
      * Maybe a video?
      * Photos?
      * Current popular bikes? Products?
* Login/Register
  + Login form
    - Or simple page, with just login form
      * For demo purposes all users can login as either Customer or Employee
        + Just have two buttons for login to choose what side they see
  + Registration form
    - Similar to login, but more fields
      * Phone, email, address, other thinks that user would need in database
* Customer (attractive UI for an enjoyable experience)
  + Make sure customer side css works on phones as well
    - Potentially make it a phone app?
  + Header
    - Site name/logo
    - Nav bar
      * Home
      * Tickets
      * Shopping
    - Cart/checkout
  + Main Home Page
    - Dashboard style
      * Brief overview of current tickets
        + Ability to click on ticket and view details
      * Some suggested products
        + Customer can click on products to view details
        + Also includes a search bar for detail search

Will navigate to shopping page

* + Ticket Scheduling Page
    - Customer can schedule a new ticket
      * Utilize same schedule that employees have access to
      * Customer enters their information and some details on what they would like done
      * Ticket gets submitted for approval
      * On Employee side the ticket is reviewed and confirmed or the customer is contacted if there are issues.
  + Ticket Details Page
    - Display selected ticket details
      * If ticket is complete and ready for payment, customer has option to pay online
      * Customer can view comments and respond but not edit techs notes
  + Shopping Page
    - Featured search bar at top center
      * Add a dropdown to pick categories
    - Initially load popular/recommended products
    - Each product will be in card format with some basic information
      * Full card is clickable to navigate to detail page
      * User can also directly add to cart without navigating
    - Page scroll at top and bottom
    - Add a sort by dropdown
  + Product Detail Page
    - Display products details
    - Featured Add to Cart Button
    - Same featured search bar still at the top of the page
    - Return to results button in top left
  + Cart/Checkout Page
    - Focus on Cart
      * Edit/remove items
    - Checkout button
      * Shows checkout form
      * Various Checkout Options
        + Add to work order
        + Pickup in store
        + Delivery
    - Completed checkout navigates to home page with success modal overlay
* Employee (Functional UI for efficient navigation)
  + - make almost everything auto-save (employees can rapidly navigate without worry about loosing data
  + Header
    - Site name/logo
    - Nav bar (right)
      * Switch to Customer View
    - Ticket search (middle)
      * Dropdown to determine search type
      * Draw attention to it (convenience of searching on any screen)
  + Main Home Page
    - Dashboard Style
      * Upcoming Tickets Overview
        + Showing tickets that need to be worked on soonest
        + Show a graph of ticket completion for the day (by minutes scheduled)
        + Depending on click navigates to main ticket search page or ticket details page
      * General To Do List
        + List out shop tasks that need to be completed
        + Entirely handled on dashboard

Include add/remove/finished buttons

* + - * Schedule showing available service slots/time
        + Clicking on open slot will take you to the new Ticket page
      * Customer Ticket Review Notifications
        + When a customer creates there own ticket a notification will pop up for a tech to review/approve the ticket
        + Click on the notification to navigate to the ticket Review Page
  + Ticket Search Page
    - Main page for looking up tickets
    - Multiple search options and sort options
      * Customer
      * Bike
      * Location
      * Due date
      * status
    - Each ticket item has details for mechanic to get needed info at a glance
      * Click on item to navigate to details page
  + Create New Ticket Page
    - Page for generating and scheduling a new ticket
    - Once complete navigates to the dashboard
  + Ticket Details Page (updates)
    - Display all the details of a ticket
    - Some fields can be edited at any time
      * Internal notes
      * External notes
      * Due date
      * Status
    - Other fields must be unlocked by clicking and edit customer button
    - Bottom portion will be a search field with a display area
      * A few close matching results will be displayed here
      * Can click for more results
        + Navigates to a full search page
  + Customer Ticket Review Page
    - Same as the Ticket Details Page
    - Approve/Edit/Contact Options at the top
      * Tech reviews the information and scheduled time
      * If everything is good, then approve
      * If small edits are needed without customers approval then make changes and approve
      * If more major changes are needed, or need to be scheduled for a different day, then contact customer
  + Search Page
    - Ability to select products or labor items for searching
    - Different search categories depending on what is selected
      * Products
      * Labor
* Footer – same on every page
  + Link to project repo
  + Link to project demo video
  + Link to my linked in account
  + Link to my portfolio
* Tech Ticket Details
  + Ticket number/due date
  + Edit/delete buttons
  + Customer/bike information
  + Customer Notes
  + internal Notes
  + list of labor and parts
    - has search bar
* Tech Create New Ticket
  + - Search for
* Customer Ticket Details
  + Ticket number and Due date and status up top
  + Customer Info
  + Bike Info
  + Tech Notes
  + List of labor and parts
    - Include a total cost
    - Include a checkout button if status allows
* Customer Cart/Checkout

Project Minimum Requirements (9 will pass)

* Database
  + Has at least 2 tables
  + Contains at least 1 one:many relationship between tables
  + Tables contain data
  + Connects to server
* Server
  + Handles requests
  + Sends responses
* Front End
  + Has 3-5 interactive features
  + Communicates with server
  + Has at least 5 components
  + Stores values on state
  + Uses values from state to display information and/or make requests
  + Uses props to pass data and/or methods
  + Includes custom styling