Project Documentation

# Project Details

|  |  |
| --- | --- |
| **TITLE** | Trade Tracking Website – TradeLab V1 |
| **CLIENT** | Martz Industries |
| **DEADLINE** | 07/01/2025 |
| **DESIGNER/DEVELOPER** | Tony Martz |

# Project Analysis

## Scope

a. The work required to complete the project includes:

* Researching and compiling the most important statistics and metrics (including their formulas) for traders, as well as the best methods to display their data in graphs and charts.
* Designing and developing a user-friendly, easy-to-use, intuitive website for users to input, track and analyze their trades.
* Creating a free platform with the most important features to traders.
* Ensuring the website is easy to navigate and accessible.
* Creating a consistent design and layout across all pages.
* Database integration for users to store their log in, and trade data.
* Fully secure tool integration to protect user data.
* Login and authentication features with fully secure servers.
* Main goal – complete a trade entry form with backend for users to enter and store trade data.

b. An action that would be considered out of scope is:

* Developing a chart to display metrics.

## Deliverables

The deliverables for this project will include:

* All HTML files for the web pages (trade entry, import, all trades, analytics, market insights, calendar, settings, user guide, privacy policy, risk disclaimer notice).
* CSS stylesheet(s) for styling the web pages.
* Required JavaScript files for this sprint (global, sidebar, trade entry, API, server, database)
* Database files and schemas.
* Any necessary assets, such as font files.
* Technology Stack (detailed):
  + Frontend: HTML5, CSS3, JavaScript, Chart.js, Sheets.js.
  + Backend: Node.js, Express, REST
  + Database: SQLite3 (prototyping), PostgreSQL
  + API communication: Axios for HTTP requests.

**Assumptions**

* Manual trade entry will be sufficient for the MVP.
* The initial focus is on individual traders, not institutional clients or teams.
* We will not be integrating with any external brokerage APIs in the MVP.
* Basic security measures will be implemented for the MVP, but comprehensive security audits will be a later phase.
* Basic user authentication (login/registration) will be included in the MVP.

**Success Metrics**

* Number of registered users.
* Number of trades entered into the system.
* User engagement (e.g., time spent on the platform, number of pages visited).
* Qualitative feedback from initial users.

**Project Risks and Mitigation**

* **Risk:** Scope creep.
  + **Mitigation:** Strict adherence to the MVP feature set and a well-defined backlog for future iterations.
* **Risk:** Performance issues with a growing number of trades.
  + **Mitigation:** Use SQLite for efficient data storage and retrieval in the MVP.
* **Risk:** Usability issues with the initial design.
  + **Mitigation:** Prioritize a clean and intuitive UI based on user personas and conduct basic usability testing.
* **Risk:** Data loss.
  + **Mitigation:** Implement regular database backups.

**High-Level Architecture**

The TradeLab V1 application will follow a client-server architecture:

1. **Frontend:** The user interacts with the HTML, CSS, and JavaScript in their web browser. This frontend sends requests to the backend API.
2. **Backend:** The Node.js/Express.js API receives requests from the frontend, processes them (e.g., saving data, retrieving data), and sends responses back to the frontend.
3. **Database:** SQLite stores the application's data, including user information and trade details.

**Definition of Done (DoD) for the MVP**

* Users can successfully register, log in, and log out.
* Users can manually input trade data with all required fields.
* Trade data is persistently stored in the database.
* Users can view a list of their entered trades.
* Basic styling is applied for readability and usability.
* Basic error handling is implemented for user input.
* The application is deployed to a testable environment.

**Sprint Planning Approach**

We will be using an Agile/Scrum approach with short sprints (1-2 weeks), daily stand-up meetings, and iterative development based on user feedback. Each sprint will focus on delivering a specific set of features, with regular reviews and retrospectives to ensure continuous improvement.

## Audience

a. The audience for the end product is:

* Individual traders looking for a free and easy-to-use platform to track and analyze their personal trading performance across various asset classes. This includes both novice and experienced traders who want to gain insights into their trading habits and identify areas for improvement.

b. Their needs and how the project will meet them:

* **Need 1: A platform with free basic features.** Traders need a platform that offers the most critical features for free.
  + How the project meets the need: The project will provide a platform where such features will be offered at no cost, only charging for non-essential, nice-to-have features.
* **Need 2: An intuitive platform.**  Traders need a platform that is easy to use and where navigation and menus are easy to find.
* How the project meets the need: The project will use the most recent standards and practices to create a website easy to navigate for the user, including the use of appropriately sized fonts and components and user-friendly layout.
* **Need 3: A trade entry form to enter trade data.** Traders need an intuitive, reliable form to enter their trade data that will store it for later analysis.
  + How the project meets the need: The project will use a consistent, intuitive layout, easy to find menus, and intuitive buttons to allow users easy data input.

# Project Plan

## Task List

**Planning**

* Define project scope and deliverables.
* Identify target audience and their needs.
* Create a project timeline and milestones.
* Develop an asset list.
* Outline page content and layout.
* Define database schema.
* Define security tools needed for a secure site.
* Identify best database service.
* Conduct competitive analysis
* Define user flows
* Select project management tools

**Design**

* Design user experience (navigation, scrolling, page transitions, information scanning).
* Create a persona for the target audience.
* Design the page UI layout.
* Design the component UI layout (HTML elements, CSS properties, form).
* Design database schema.
* Design data flow.
* Design data objects: Trade, User, Accounts
* Develop a style guide

**Development**

* Create HTML files for each page.
* Develop CSS stylesheets for styling.
* Implement navigation and page structure.
* Ensure cross-browser compatibility.
* Write necessary JavaScript files for menu, sidebar, and login functionality.
* Create database schema.
* Create critical database automation files such as: migrations, data integrity, data validation.
* Create trade entry form.
* Implement backend API endpoints
* Connect frontend to backend
* Implement user authentication
* Write unit tests for backend and frontend

**Testing**

* Test website functionality and usability.
* Validate HTML and CSS code.
* Check for accessibility issues.
* Test on different devices and screen sizes.
* Ensure trade form performs as expected.
* Ensure data validation occurs properly.
* Test data storage.
* Test site security.
* Perform integration testing
* Conduct performance testing
* Perform security testing
* User acceptance testing

**Deployment**

* Prepare files for deployment.
* Upload files to the server.
* Configure website settings.
* Set up a staging environment
* Perform final testing in the staging environment
* Set up monitoring and logging
* Create a deployment rollback plan

## Timeline

* Planning: 2 days
* Design: 2 days
* Development: 7 days
  + Sprint 1: User authentication and trade entry form (3 days)
  + Sprint 2: Trade display and basic data retrieval (2 days)
  + Sprint 3: Database integration and API endpoints (2 days)
* Testing: 3 days
* Deployment: 1 day

## Milestone

* Milestone 1: Project plan and persona completed.
* Milestone 2: UI design (page and component layouts) approved.
* Milestone 3: Core trade entry functionality complete (HTML, CSS, Backend)
* Milestone 4: Basic trade display and charting complete.
* Milestone 5: HTML and CSS development completed.
* Milestone 6: JavaScript files completed.
* Milestone 7: Website testing completed.
* Milestone 8: Website is deployed.

## Asset list

File Name: index.html

* File Path: /
* Description: Landing page and CTA

File Name: auth.html

* File Path: /
* Description: Login page, redirected here from any page outside dashboard (ex, landing page, contact, user guide)

File Name: import.html

* File Path: /pages/
* Description: Page used to import trades using excel sheets.

File Name: trade-entry.html

* File Path: /pages
* Description: Page with form to enter trade data.

File Name: contact.html

* File Path: /pages
* Description: HTML page for the information request form, feedback submission, support request.

File Name: styles.css

* File Path: /css/
* Description: CSS stylesheets for all pages

Description: Main sans-serif font (Space Mono)

Description: Heading sans-serif font (Roboto Condensed)

Description: Monospace font (Roboto Mono)

Description: Main background color: #f8f9fa

Description: Dark background color: #121620

Description: Light text color: #e2e8f0

Description: Main text color: #1a2942

Description: Card background color: #fff

Description: Border color: #e2e8f0

Description: Hover background color: #f0f4f8

Description: Shadow color: 0 2px 4px rgba(0, 0, 0, 0.5)

Description: Accent primary color: #002FA7

Description: Accent primary-light color: #1144BD

Description: Accent secondary color: #A77800

Description: Accent secondary-light color: #C68E00

Description: Sidebar color: #ffffff

Description: Sidebar hover color: #f0f4f8

Description: Sidebar active color: #e5eeff

Description: Notification success color: #2E7D32

Description: Notification fail/error color: #C62828

Description: Footer color: #1a2942

Image assets: Logo.png, favicon.ico

* File Path: /images/
* Description: Logo image and Favicon

Third-party libraries: Chart.js

* File Path: /js/lib/chart.js
* Description: Charting library

Database files: trade\_data.sqlite

* File Path: /db/trade\_data.sqlite
* Description: SQLite database file

Configuration files: config.js

* File Path: /config/config.js
* Description: Application configuration file

File Path: /

File Path: /

File Path:

## Page Content and Verbiage

* **Landing Page:**
  + **Heading: TradeLab**
  + **Sections: Hero, Features, Pricing, About, CTA**
  + **Content: Landing page, with information about services, features, pricing, and Sign up.**
* **Contact Page:**
  + **Heading: Contact Us**
  + **Fields: Name, Email, Subject, Message**
  + **Button: Send Message**
  + **Contact Info: Hours, email**
  + **Verbiage: “Have questions, suggestions, or need support? We're here to help! Fill out the form below and we'll get back to you as soon as possible.”**
* **Auth Page:**
  + **Heading: Login/Signup**
  + **Fields: Email, password**
  + **Additional: “Remember Me” checkbox, Google/email login**
  + **Button: Login/Signup**
* **Trade Entry Page:**
  + **Heading: Trade Entry**
  + **Sections: Cumulative PnL, Today’s PnL, Trade Entry form, Recent Trades table**
    - **Fields: Ticker (string), Quantity (number), Entry Price (number), Long/Short (boolean), Exit Price (number), Market (string), Date (date), Notes (string)**
  + **Buttons: Long/Short, Add Trade**
* **Footer: TradeLab, Quick Links, Legal, Social**

**Metadata**

* **Viewport meta tag for responsiveness.**
* **Character set meta tag (UTF-8).**
* **Title for each page (e.g., "HTML Tags Cheat Sheet," "CSS Box Model Cheat Sheet").**
* **Description meta tag for each page.**
* **Author meta tag.**
* **Social media meta tags (Open Graph or Twitter Card) - If used, otherwise, omit.**

**Project Design**

**User Experience Plan**

* **Navigation Menu: The navigation menu will appear at the top of each page. It will be a horizontal bar with links to the main sections (Home, About, Pricing, Features, Contact, Dashboard).**
  + **Hover: Links will change color and/or underline on hover.**
  + **Click: Links will navigate the user to the corresponding page.**
  + **Focus: Links will have a visible focus outline for keyboard navigation.**
* **Scrolling:**
  + **The body will have hidden overflow, and the main content area will have auto overflow, allowing for vertical scrolling of the main content.**
  + **The navigation will remain fixed at the top of the page.**
  + **The footer will be at the bottom of the page.**
* **Page Transitions: Users will change pages by clicking on the navigation links. The menu will always be visible at the top of the page, providing constant access to other sections.**
* **Information Scanning:**
  + **The pages will use clear headings, subheadings, lists, and tables to organize information.**
  + **A consistent layout will be used across all pages to help users quickly find the information they need.**
  + **Important buttons or links, will be styled with an accent color to catch the reader’s attention.**
* **Error Handling: Form validation errors will be displayed inline, near the corresponding input field, with a clear error message.**
* **Feedback Mechanisms:**
  + **Successful trade addition: A success message will be displayed to the user (e.g., a green notification).**
  + **Form submission: A loading indicator will be displayed while the data is being processed.**
* **Accessibility:**
  + **Semantic HTML will be used to ensure proper structure and meaning.**
  + **Sufficient color contrast will be used for all text and interactive elements.**
  + **All images will have descriptive alt attributes.**
  + **Form labels will be properly associated with their input fields.**
  + **The site will be navigable using only a keyboard.**
  + **ARIA roles will be used where necessary to enhance screen reader compatibility.**
* **Home Page Greeting: There isn't a traditional "home page" in this project. The landing page will be an information page with the features and services for the platform with a CTA.**

|  |  |  |
| --- | --- | --- |
| Female Profile with solid fill | Alex “The Beginner Trader”Description Alex is an individual investor who has been actively trading stocks and occasionally options for the past three years. They work full time in a tech-related field but dedicate several hours each week to trading early morning. Alex is a self-taught in trading, having learned through online courses, books, and following market news. | Demographics **Age** 35  **Gender** Male  **Marital Status** Single  **Occupation** Full-time Engineer |

## Project Persona

**Page UI Design**

**The overall layout of the pages will consist of the following sections:**

1. **Header: Contains top navigation menu, and page header.**
2. **Navigation: A horizontal menu with links to the main site pages (ex, index, contact, legal pages).**
3. **Sidebar (While logged in): A vertical side menu with links to the different pages in a user dashboard.**
4. **Main Content: The main area of the page, containing content based on page.**
5. **Footer: Contains the site name, footer navigation, legal disclosure, and social links.**

**CSS Layout Method:**

* **I will use CSS FlexBox for the Top Menu and Sidebar as well as the footer. The main content will use a combination of either FlexBox, Grid, or both.**

**Component UI Design**

* **Buttons:**
  + **Style: Color: Primary accent color with hover and active states, Padding: 8px 16px, Border: None, Rounded corners, Shadow: small shadow.**
  + **Behavior: Clickable, provides visual feedback on interaction (e.g., slight color change, shadow).**
* **Input fields:**
  + **Style: Borders: 1px solid border color, Focus effects: border color change, shadow, Labels: Placed above the input field.**
  + **Behavior: Displays user input, provides visual feedback on focus, displays validation messages.**
* **Tables:**
  + **Style: Headers: Bold text, alternating row colors for readability, Borders: thin borders.**
  + **Behavior: Responsive, supports sorting (future), displays data in a structured format.**
* **Forms:**
  + **Layout: Labels above input fields, consistent spacing, clear grouping of related fields.**
  + **Error message display: Inline, near the invalid field, using a red color and an error icon.**
  + **Button placement: At the bottom of the form, aligned to the right.**
* **Cards:**
  + **Style: Borders: 1px solid border color, Shadow: small shadow, Background: Card background color.**
  + **Content structure: Heading, optional subheading, body text, optional footer.**
* **Alerts/Notifications:**
  + **Types: Success (green), Error (red), Warning (yellow).**
  + **Style: Background color, icon, clear message.**
  + **Behavior: Display duration: 3-5 seconds, Placement: Top of the page, below the header, or as a toast.**