

MDS Stock Exchange Developer Documentation

Author: Veljko Matić

Belgrade, February 2025.

Table of Contents

1. Introduction.....	1
2. Project Structure.....	2
2.1 HTML Structure (index.html).....	2
2.2.CSS Styles (assets/css/styles.css)	3
2.2 JavaScript Scripts	5
2.2.1 script.js - Navigation and Interaction	5
2.2.2 stock-kajron.js - Stock Price Display.....	7
2.2.3 custom-chart.js - Chart Display	9
2.2.4 policy.js - Privacy Policy Modal	10
3. How to Maintain Web Site.....	11
4. Future Development	12
5. Conclusion	13
6. References	14

1. Introduction

MDS Stock Exchange is an innovative web application that provides users with the ability to invest and educate themselves about the capital market.

The task was to create a Single Page presentation website for the MDS platform, featuring a modern and futuristic design that reflects the spirit of the contemporary capital market. This website will serve as a key information hub for potential users and investors about our project. The goal is to present MDS stock and our exchange to users in a simple and visually appealing way.

The website should be intuitive and modern, with a clear hierarchy of information that helps users easily understand the key details about the project and the services the platform offers.

Fundamental technical requirements are:

- Use the latest HTML5 and CSS3 standards for the structure and styling of the page.
- JavaScript or a JS framework.
- Optimize images and other media for fast page loading to ensure the best possible user experience.

The purpose of this document is to provide detailed technical information on the structure, functionalities, and usage of the platform.

2. Project Structure

2.1 HTML Structure (index.html)

The webpage is divided into several main sections, with modern HTML5 elements being used for better semantics and structure:

- **Navigation (<nav>)** (HTML5 feature: Semantic navigation element)
 - Logo with links for language switching
 - Main menu with items: *Home, About Us, Our Services, Contact*
- **Ticker Section (<div class="ticker-container">)** (HTML5 feature: Use of <div> for dynamic content updates)
 - Dynamic display of market data
- **Hero Section (<header id="home">)** (HTML5 feature: Header element for defining introductory content)
 - Title and call-to-action with a subscription form for notifications
- **About Us (<section id="about">)** (HTML5 feature: Section element to structure page content)
 - Textual information about the platform
 - Dynamic chart displaying stock growth
- **Services Section (<section id="services">)** (HTML5 feature: Section element for better content organization)
 - Cards displaying different service offerings
- **Contact Section (<section id="contact">)** (HTML5 feature: Use of <form> with new input types)
 - Contact form using:

```
<form id="contact-form">
  <input type="text" name="name" placeholder="Your Name" required>
  <input type="email" name="email" placeholder="Your Email" required>
  <textarea name="message" placeholder="Your Message" required></textarea>
  <button type="submit">Send</button>
</form>
```

Social media links using:

```
<a href="https://www.facebook.com/">Facebook</a>
<a href="https://www.instagram.com/">Instagram</a>
<a href="https://www.linkedin.com/">LinkedIn</a>
```

- **Footer (<footer>)** (HTML5 feature: Footer element for page-wide metadata and links)

Contains legal disclaimers and social media links

2.2. CSS Styles (assets/css/styles.css)

CSS deployment characteristics are:

- Main styles for typography, colors, and element layout
- Navigation: Styled menu with *hover* effects
- Service Cards: Animations and expanded descriptions
- Responsiveness: Adaptation for mobile devices

Modern CSS3 techniques have been used to enhance the styling and interactivity of the platform:

- **Navigation (nav)** (CSS3 feature: *box-shadow, transition effects*)

```
nav {
  box-shadow: 0 4px 10px rgba(0, 0, 0, 0.2);
  transition: background-color 0.3s ease-in-out;
}
```

- **Interactive Logo (nav .logo img)** (CSS3 feature: *transform effects*)

```
nav .logo img {
  transition: transform 0.3s ease-in-out;
}
nav .logo img:hover {
  transform: rotate(10deg) scale(1.1);
}
```

- **Ticker Animation (.ticker)** (CSS3 feature: *@keyframes animation*)

```
@keyframes scroll {
  0% { transform: translateX(100%); }
  100% { transform: translateX(-100%); }
}
.ticker {
  animation: scroll 30s linear infinite;
}
```

- **Hero Section (header)** (CSS3 feature: *background-size*, *flexbox* for layout)

```
header {
  background-size: cover;
  display: flex;
  flex-direction: column;
  align-items: center;
}
```

- **Call-to-Action Form (.cta-form)** (CSS3 feature: *box-shadow*, *border-radius*)

```
.cta-form {
  box-shadow: 0 4px 10px rgba(0, 0, 0, 0.1);
  border-radius: 10px;
}
```

- **Contact Form (#contact form)** (CSS3 feature: *transition*, *focus effects*)

```
#contact form input:focus {
  border-color: rgb(179, 41, 41);
  background-color: #fff;
}
```

2.2 JavaScript Scripts

2.2.1 script.js - Navigation and Interaction

- Implements smooth scrolling to webpage sections.
- Enables displaying and hiding additional service details

```
document.addEventListener('DOMContentLoaded', () => {
  const links = document.querySelectorAll('.scroll-to');

  links.forEach(link => {
    link.addEventListener('click', (event) => {
      event.preventDefault();

      const targetId = link.getAttribute('href').substring(1);
      const targetElement = document.getElementById(targetId);

      if (!targetElement) {
        return;
      }

      const navbarHeight = document.querySelector('nav').offsetHeight;

      window.scrollTo({
        top: targetElement.offsetTop - navbarHeight,
        behavior: 'smooth'
      });
    });
  });

  document.addEventListener('DOMContentLoaded', () => {
    const buttons = document.querySelectorAll('.read-more-btn');
    const servicesSection = document.getElementById('services');
    const image = document.querySelector('.services-image img');
    const navbar = document.querySelector('nav');

    function toggleDescription(button) {
      const card = button.closest('.service-card');
      const fullDescription = card.querySelector('.full-description');
      const isExpanded = fullDescription.style.display === 'block';
    }
  });
});
```

```
if (isExpanded) {
  fullDescription.style.display = 'none';
  button.textContent = 'Pročitaj više';

  servicesSection.scrollToView({
    behavior: 'smooth',
    block: 'start'
  });
  window.scrollTo({
    top: servicesSection.offsetTop - navbar.offsetHeight,
    behavior: 'smooth'
  });
} else {
  document.querySelectorAll('.full-description').forEach(desc => {
    desc.style.display = 'none';
  });
  document.querySelectorAll('.read-more-btn').forEach(btn => {
    btn.textContent = 'Pročitaj više';
  });

  fullDescription.style.display = 'block';
  button.textContent = 'Sakrij';

  fullDescription.scrollToView({
    behavior: 'smooth',
    block: 'center'
  });
}

buttons.forEach(button => {
  button.addEventListener('click', () => {
    toggleDescription(button);
  });
});
});
```


2.2.2 stock-kajron.js - Stock Price Display

- Calls an API to update stock prices.
- Displays real-time data in the ticker section.

```
const API_KEY = "LFLW8PXLZXR2023S";
const symbols = ["AAPL", "GOOGL", "MSFT", "TSLA", "AMZN"];

let lastFetchTime = 0;
let stockData = {};

const defaultData = {
  AAPL: { price: 227.63, color: "white", arrow: "" },
  GOOGL: { price: 185.34, color: "red", arrow: "↓" },
  MSFT: { price: 409.75, color: "white", arrow: "" },
  TSLA: { price: 361.62, color: "green", arrow: "↑" },
  AMZN: { price: 229.15, color: "red", arrow: "↓" }
};

async function fetchStockData() {
  const currentTime = Date.now();

  if (currentTime - lastFetchTime > 3600000) {
    try {
      let tickerContent = '';
      for (let i = 0; i < symbols.length; i++) {
        const symbol = symbols[i];
        const response = await fetch(
          `https://www.alphavantage.co/query?function=GLOBAL_QUOTE&symbol=${symbol}&apikey=${API_KEY}`
        );
        const data = await response.json();
        const stockInfo = data["Global Quote"];

        if (stockInfo) {
          const price = parseFloat(stockInfo["05. price"]);
          const changeValue = parseFloat(stockInfo["09. change"]);

          let color = "white";
          let arrow = "";

          if (changeValue > 0) {
            color = "green";
            arrow = "↑";
          } else if (changeValue < 0) {
            color = "red";
            arrow = "↓";
          }

          stockData[symbol] = { price, changeValue, color, arrow };

          tickerContent += `
            <span style="color: ${color};">${symbol}: ${price}</span>
            <span>${arrow}</span> |
          `;
        } else {
          tickerContent += `
            <span style="color: ${defaultData[symbol].color};">${symbol}: ${defaultData[symbol].price}</span>
            <span>${defaultData[symbol].arrow}</span> |
          `;
        }
      }
    } catch (error) {
      console.error("Error fetching stock data:", error);
    }
  }
}
```

```

        tickerContent += tickerContent;
        tickerContent += tickerContent;

        document.getElementById("stock-ticker").innerHTML = tickerContent;
        lastFetchTime = currentTime;
    } catch (error) {
        console.error("Greška pri dohvatanju podataka:", error);
        let tickerContent = '';
        for (let i = 0; i < symbols.length; i++) {
            const symbol = symbols[i];
            tickerContent += `
                <span style="color: ${defaultData[symbol].color};">${symbol}: ${defaultData[symbol].price}</span>
                <span>${defaultData[symbol].arrow}</span> |
            `;
        }
        document.getElementById("stock-ticker").innerHTML = tickerContent;
    }
} else {
    let tickerContent = '';
    for (let i = 0; i < symbols.length; i++) {
        const symbol = symbols[i];
        const data = stockData[symbol] || defaultData[symbol];

        tickerContent += `
            <span style="color: ${data.color};">${symbol}: ${data.price}</span>
            <span>${data.arrow}</span> |
        `;
    }
    document.getElementById("stock-ticker").innerHTML = tickerContent;
}
}

fetchStockData();

setInterval(fetchStockData, 86400000);

```

2.2.3 custom-chart.js - Chart Display

- Uses the *Chart.js* library to visualize stock growth.
- Defines data and chart display options.

```
const ctx = document.getElementById('myChart').getContext('2d');

const myChart = new Chart(ctx, {
  type: 'line',
  data: {
    labels: ['Jan', 'Feb', 'Mar', 'Apr', 'May', 'Jun', 'Jul', 'Aug', 'Sep', 'Oct', 'Nov', 'Dec'],
    datasets: [{
      data: [100, 120, 150, 170, 200, 220, 260, 300, 350, 400, 450, 500],
      borderColor: 'ff0000',
      backgroundColor: 'rgba(255, 0, 0, 0.1)',
      tension: 0.4,
      fill: true,
      borderWidth: 3,
      pointBackgroundColor: 'ff0000',
      pointRadius: 5,
      pointHoverRadius: 7
    }]
  },
  options: {
    responsive: true,
    scales: {
      x: {
        grid: {
          color: '#333',
        },
        ticks: {
          color: 'fff',
        }
      },
      y: {
        grid: {
          color: '#333',
        },
        ticks: {
          color: 'fff',
          beginAtZero: true,
          callback: function(value) {
            return value + '$';
          }
        }
      }
    },
    plugins: {
      legend: {
        display: false
      }
    }
  }
});
```

2.2.4 policy.js - Privacy Policy Modal

- Enables opening and closing of the modal window containing privacy policy information.

```
document.getElementById("privacy-link").addEventListener("click", function(event) {
    event.preventDefault();
    document.getElementById("privacy-modal").classList.add("show");
});

document.getElementById("close-btn").addEventListener("click", function() {
    document.getElementById("privacy-modal").classList.remove("show");
});

window.addEventListener("click", function(event) {
    if (event.target === document.getElementById("privacy-modal")) {
        document.getElementById("privacy-modal").classList.remove("show");
    }
});
```

3. How to Maintain Web Site

1. Starting the Project:

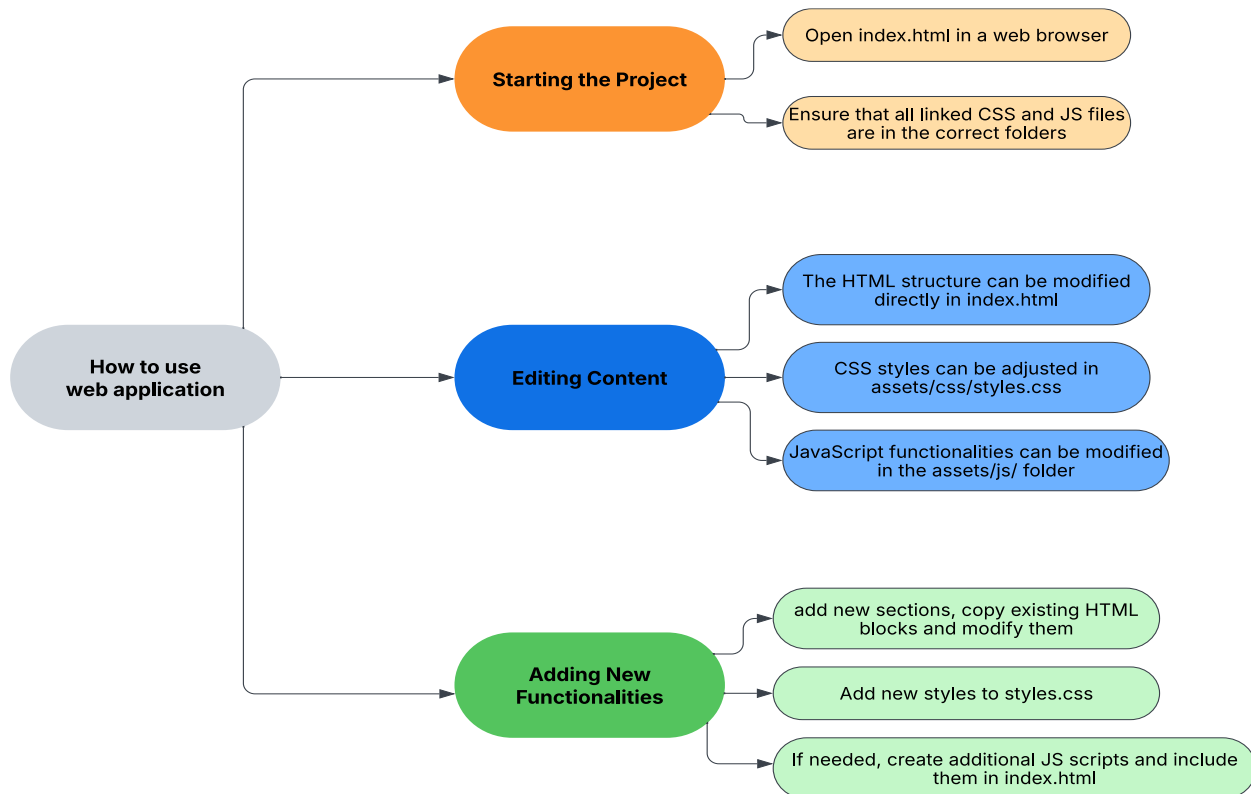
- Open index.html in a web browser.
- Ensure that all linked CSS and JS files are in the correct folders.

2. Editing Content:

- The HTML structure can be modified directly in index.html.
- CSS styles can be adjusted in assets/css/styles.css.
- JavaScript functionalities can be modified in the assets/js/ folder.

3. Adding New Functionalities:

- To add new sections, copy existing HTML blocks and modify them.
- Add new styles to styles.css.
- If needed, create additional JS scripts and include them in index.html.



4. Future Development

Future improvements for a static website can enhance performance, usability, and functionality. Here are some potential improvements:

Interactivity & User Experience:

- Progressive Web App (PWA) – Enable offline access and installable capabilities.
- Integrate an API for real stock market data in the ticker section
- Enhanced Animations – Use CSS animations or lightweight JavaScript libraries (GSAP, AOS).
- Dark Mode – Provide a user-friendly dark/light mode toggle.
- Accessibility Improvements – Implement ARIA roles and ensure screen reader compatibility.

SEO & Analytics:

- Structured Data Markup – Improve SEO with schema.org tags.
- Improved Metadata – Use proper title, description, and Open Graph tags for better search visibility.
- Analytics Integration – Add Google Analytics or alternative tools for tracking user behavior.

Content Management & Automation:

- Headless CMS Integration – Use a CMS like Contentful or Strapi for easier content updates.
- Automated Deployments – Implement CI/CD pipelines for efficient updates.
- Static Site Generators – Convert to a framework like Next.js, Hugo, or Jekyll for better scalability.

Security Enhancements:

- Enable HTTPS – Ensure SSL/TLS encryption is active.
- CSP & Security Headers – Prevent XSS and other vulnerabilities.
- Regular Updates – Keep dependencies updated if using external libraries.

5. Conclusion

MDS Stock Exchange has been developed as a modern and dynamic web application that enables users to make informed investments. This documentation provides all necessary details to maintain and further enhance the platform.

Future improvements could focus on expanding backend functionality to include secure user authentication and advanced data processing. Additionally, integrating real-time stock market APIs would make the platform even more robust and informative. Enhancing the UI with modern CSS3 animations and interactive elements will further improve the overall user experience.

Continuous development, performance optimization, and security enhancements will ensure that *MDS Stock Exchange* remains a competitive and valuable tool in the financial technology sector. By maintaining clean and structured code, adhering to modern web standards, and listening to user feedback, the platform can evolve into an industry-leading solution.

6. References

Here is the short list of referenced documentation and links that are used:

External Libraries and Services:

- Chart.js – A simple yet flexible JavaScript charting library for the modern web.
[Chart.js](#)
- Font Awesome Icons – A toolkit for icons and fonts, widely used by designers, developers, and content creators.
[Font Awesome](#)
- Google Fonts – A collection of free and open-source fonts, including 'Signika Negative'.
[Google Fonts](#)
- Alpha Vantage API – Provides free JSON APIs for real-time and historical stock market data, options data, and more.
[Alpha Vantage](#)

Internal Project Resources:

JavaScript Files:

- assets/js/script.js – Handles smooth scrolling and expands service descriptions.
- assets/js/stock-ticker.js – Updates the stock ticker using the Alpha Vantage API.
- assets/js/custom-chart.js – Creates a line chart of stock growth using Chart.js.
- assets/js/policy.js – Controls the display and closure of the privacy policy modal.

CSS Files:

- assets/css/styles.css – Contains the main styles for the website.

Images and Icons:

- assets/images/MDS.png – Logo and illustrations.
- assets/images/MDS_LOGO.svg – SVG version of the logo.