Shri Vile Parle Kelavani Mandal's
DWARKADAS J. SANGHVI COLLEGE OF ENGINEERING
(Autonomous College Affiliated to the University of Mumbai)

# Fund - Visor (Mutual Fund Comparison App)

Submitted in partial fulfillment of the requirement
of the degree of B.Tech Electronics

**Bachelor of Technology in**

**Electronics Engineering**

By

**Vaibhavi Gandhi** **60001200071**
**Ronak V. Kulkarni** **60001200076**

Under the guidance of

**Prof. Rashmi Ravikumar**

University of Mumbai

**A.Y. 2023 – 2024**

Shri Vile Parle Kelavani Mandal's
DWARKADAS J. SANGHVI COLLEGE OF ENGINEERING
(Autonomous College Affiliated to the University of Mumbai)

# CERTIFICATE

This is to certify that the project entitled, **"Fund - Visor (Mutual Fund Comparison App)"** is a bonafide work of **"Vaibhavi Gandhi" (60001200071), "Ronak V. Kulkarni" (60001200076)** submitted in the partial fulfillment of the requirement for the award of the Bachelor of Technology in Electronics Engineering

Prof. Rashmi Ravikumar
**Name of the Guide**

Dr. Prasad Joshi
**Head of the Department**

Dr. Hari Vasudevan
**Principal**

**Place:**

**Date:**

Shri Vile Parle Kelavani Mandal's
DWARKADAS J. SANGHVI COLLEGE OF ENGINEERING
(Autonomous College Affiliated to the University of Mumbai)

# DECLARATION

We declare that this written submission represents our ideas in our own words and where others ideas or words have been included, we have adequately cited and referenced the original sources. We also declare that, we have adhered to all the principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in our submission. We understand that any violation of the above will be a cause for disciplinary action by the Institute and can also evoke penal action from the sources, which have thus not been properly cited or from whom proper permission has not been taken, when needed.

**Vaibhavi Gandhi (60001200071) (Sign against the name)**

**Ronak V. Kulkarni (60001200076) (Sign against the name)**

**Place:**

**Date:**

Shri Vile Parle Kelavani Mandal's
DWARKADAS J. SANGHVI COLLEGE OF ENGINEERING
(Autonomous College Affiliated to the University of Mumbai)

# APPROVAL SHEET

Project entitled, **"Fund - Visor (Mutual Fund Comparison App)"**, submitted by **"Vaibhavi Gandhi" (60001200071), "Ronak V. Kulkarni" (60001200076)** is approved for the award of the Bachelor of Technology in Electronics Engineering

**Signature of Internal Examiner** **Signature of External Examiner**

**Name of the Guide**
Prof. Rashmi Ravikumar

**Head of the Department**
Dr. Prasad Joshi

**Principal**
Dr. Hari Vasudevan

**Place:**

**Date:**

# ABSTRACT

The "Fund Visor" project is a web application that is centered on the user and aims to assist investors in making well-informed decisions on their investments in mutual funds. In order to meet the requirements of users who are not familiar with the complexities of the stock market, the program provides a safe login with an easy-to-use dashboard that includes three blank cards for the purpose of making hypothetical comparisons between mutual funds. In order to fulfill a critical requirement, Fund Visor provides users with the ability to intelligently diversify their investments. This is especially helpful for individuals who rely on professional Fund Managers at banks. A detailed table that includes serial numbers, stock names, and percentage allocations is provided by the application's primary function, which is a side-by-side comparison of selected mutual funds. Fund Visor gives users the ability to make well-informed selections, which improves overall investing strategies. It does this by assisting users in avoiding successive funds that have stock compositions that are similar to one another. In addition to its role as a mutual fund advisor, this project also plays an important role as a beneficial tool that assists users in making investment decisions that are diverse and intelligent, thereby contributing to their overall financial well-being.

# ACKNOWLEDGEMENTS

This The making of project involves teamwork, vision, patience and perseverance on the part of group members. But a team can achieve a greater height only by proper guidance from faculty members and friends. Hence we would like to express our gratitude to all those who in instrumental during the course of developing of the project.

First we would like to thank our internal guide Prof. Rashmi Ravikumar for her valuable inputs and timely guidance. She always had very practical suggestions and sound knowledge.

We would also like to thank our Laboratory-Assistants and Laboratory-Attendants for the help and co-operation offered by them along every stage of the project.

Lastly, we would like to thank our head of department, Dr. Prasad S. Joshi for extending his support towards the completion of this project and for the timely encouragement.

---

(Signature)

# Table of Contents

# List of Figures

# Chapter 1

# Introduction

## 1.1 Description :

In today's dynamic financial landscape, mutual funds stand as one of the most popular investment avenues, offering diversification, professional management, and accessibility to investors. However, with a plethora of options available, selecting the right mutual fund can be a daunting task. To address this challenge, our web application aims to empower users with comprehensive tools and insights to make informed investment decisions.

Upon accessing the application, users are greeted with a user-friendly interface featuring a secure login page. Once authenticated, users are directed to a dashboard comprising three essential tabs: Compare Mutual Funds, SIP Calculator, and LumpSum Calculator.

The flagship feature of our application resides within the "Compare Mutual Funds" tab. Here, users can leverage a powerful search bar to explore and add mutual funds of interest. Upon selecting a mutual fund, detailed information regarding its performance, investment strategy, expense ratio, and historical returns is presented. Crucially, our application distinguishes itself by offering a unique capability to compare the top 10 holdings of multiple mutual funds simultaneously. This comparative analysis empowers users to discern key differences and similarities between funds, aiding in the selection of the most suitable investment options tailored to their financial objectives and risk tolerance.

Additionally, our application provides supplementary tools such as the SIP (Systematic Investment Plan) Calculator and LumpSum Calculator. These calculators enable users to estimate potential returns and plan their investments efficiently, whether through regular contributions or lump-sum investments.

By amalgamating cutting-edge technology with robust financial analytics, our web application endeavors to streamline the mutual fund selection process, demystify investment jargon, and empower users to embark on their wealth-building journey with confidence and clarity.

# 1.2 Overview :

## 1.2.1 Login / Authentication:

The Login / Authentication feature serves as the virtual gateway to your platform, ensuring that users' interactions with your web application are not only secure but also seamlessly personalized. Through a meticulously crafted authentication process, users are granted access only after providing their unique credentials, fostering a sense of trust and confidence in the platform's security measures. This mechanism acts as a formidable barrier against unauthorized access, safeguarding users' sensitive financial information and upholding their privacy rights.

In essence, the Login / Authentication feature acts as the first line of defense in safeguarding the integrity of your platform. By requiring users to authenticate themselves through a combination of username and password, the feature not only mitigates the risk of unauthorized access but also fosters a sense of accountability among users. This sense of accountability instills confidence in users, assuring them that their interactions with the platform are secure and protected.

Furthermore, the Login / Authentication feature serves as a cornerstone for implementing personalized user experiences. By authenticating users upon entry, the platform can tailor its interface and functionality to meet the unique needs and preferences of each user. This level of personalization enhances user engagement and satisfaction, fostering long-term loyalty and retention.

## 1.2.2 Lump Sum Calculator:

The Lump Sum Calculator feature emerges as a beacon of financial empowerment, guiding users through the intricate terrain of investment planning with precision and clarity. At its core, the calculator offers users a comprehensive toolset to evaluate the potential returns on lump-sum investments in mutual funds. By inputting key parameters such as the initial investment amount, anticipated rate of return, and investment horizon, users embark on a journey of exploration into the potential growth trajectory of their investments.

Through sophisticated algorithms and mathematical models, the Lump Sum Calculator unveils a panoramic view of the investment landscape, illuminating the path towards achieving users' financial aspirations. Each calculation serves as a window into the future, offering users invaluable insights into the interplay of market dynamics, compounding effects, and investment time horizons. Armed with this knowledge, users can make well-informed decisions that resonate with their long-term wealth accumulation objectives.

Moreover, the Lump Sum Calculator serves as more than just a tool for financial analysis; it is a catalyst for empowerment and informed decision-making. By providing users with a tangible representation of their investment potential, the calculator instills confidence and clarity, empowering users to take control of their financial futures with conviction and purpose.

### 1.2.3 SIP (Systematic Investment Plan) Calculator:

The SIP Calculator feature stands as a testament to the transformative power of systematic, disciplined investing. At its core, the calculator empowers users to navigate the complexities of investment planning with confidence and foresight. Through a user-friendly interface, users embark on a journey of discovery, exploring the profound impact of regular, systematic contributions to mutual funds.

By configuring variables such as the monthly investment amount, expected rate of return, and investment tenure, users embark on a virtual voyage through time, witnessing the evolution of their investments over the years. With each simulation, the calculator unravels the mysteries of compounding interest and regular contributions, empowering users to chart a course towards their financial goals with clarity and conviction.

Beyond its role as a financial tool, the SIP Calculator serves as a beacon of financial literacy and empowerment. By demystifying complex financial concepts and providing users with actionable insights, the calculator equips users with the knowledge and confidence needed to make informed investment decisions. Whether users are seasoned investors or

newcomers to the world of finance, the SIP Calculator serves as a trusted companion on their journey towards financial freedom and prosperity.

### 1.2.4 Fund Comparison :

The Fund Comparison feature represents the epitome of analytical prowess, offering users unparalleled insights into the intricacies of mutual fund investing. At its core, the feature empowers users to make informed investment decisions through comprehensive analysis and comparison of mutual funds. Through an intuitive interface, users embark on an immersive journey of exploration, delving into the nuances of various mutual funds with unparalleled depth and granularity.

Armed with a powerful search function, users navigate a vast sea of investment options, curating a bespoke selection of funds that align with their unique preferences and objectives. With each selection, users unearth a treasure trove of data, ranging from performance metrics and investment strategies to expense ratios and historical returns. However, the true brilliance of the feature lies in its ability to transcend mere data points, offering users a panoramic view of their investment landscape through comprehensive fund comparisons.

By juxtaposing the top 10 holdings of multiple funds side by side, users gain invaluable insights into asset allocation, sector exposure, and diversification benefits. This comparative analysis enables users to identify trends, spot opportunities, and construct well-balanced investment portfolios that stand the test of time. Whether users are seeking growth, income, or a combination of both, the Fund Comparison feature serves as a trusted advisor, guiding users towards their financial goals with clarity and confidence.

# Chapter 2

# Literature Review

# 2.1 Evolution and Growth of Mutual Funds in India

## 2.1.1 Historical Context and Regulatory Framework

The mutual fund industry in India has witnessed robust growth, evolving from a single entity, the Unit Trust of India (UTI) in 1964, to a diverse market with numerous players. The liberalization policies of the Indian government in the 1990s, particularly the entry of private sector mutual funds in 1993, marked a turning point for the industry. The regulatory oversight by the Securities and Exchange Board of India (SEBI) played a crucial role in fostering transparency and investor confidence [1].

Mutual funds have become an essential investment vehicle, providing individuals with an accessible means of participating in the financial markets. The concept of mutual funds dates back to the early 20th century, with the first mutual fund being established in the United States in 1924 [1]. Since then, mutual funds have evolved into a diverse and sophisticated investment option, offering a wide range of funds tailored to various investment goals and risk profiles.

## 2.1.2 Performance and Trends

The period between 1998 and 2006 saw a significant increase in the number of mutual fund schemes and assets under management (AUM). The emergence of open-end schemes, which offer greater liquidity and flexibility, became more popular compared to close-end schemes. This period also witnessed a shift in market share from public sector entities like UTI to private sector mutual funds, driven by improved performance and investor services [1].

The primary appeal of mutual funds lies in their ability to pool resources from numerous investors to create a diversified portfolio of securities. This diversification reduces the risk associated with investing in individual securities and provides access to a broader range of investment opportunities that may not be available to individual investors. Professional asset managers oversee the management of mutual funds, leveraging their expertise to make informed investment decisions and optimize returns.

The primary appeal of mutual funds lies in their ability to pool resources from numerous investors to create a diversified portfolio of securities. This diversification reduces the risk associated with investing in individual securities and provides access to a broader range of investment opportunities that may not be available to individual investors. Professional asset managers oversee the management of mutual funds, leveraging their expertise to make informed investment decisions and optimize returns.

The mutual fund industry is experiencing several notable trends that reflect the evolving landscape of financial markets and investor preferences. One significant trend is the increasing popularity of passive investing. Passive funds, such as index funds and exchange-traded funds (ETFs), have gained traction due to their lower fees and the growing awareness of the difficulty in consistently outperforming the market through active management. This shift towards passive investing is driven by a combination of cost-consciousness and the desire for simplicity and transparency in investment choices [2].

Another important trend is the integration of environmental, social, and governance (ESG) factors into investment decisions. Investors are increasingly considering ESG criteria when selecting mutual funds, driven by a growing awareness of the impact of their investments on society and the environment. Fund managers are responding to this demand by offering ESG-focused funds and incorporating ESG analysis into their investment processes [7].

The rise of digital platforms and fintech solutions is also transforming the mutual fund industry. Companies are leveraging technology to provide seamless and efficient investment experiences, offering features such as automated portfolio management, real-time performance tracking, and personalized recommendations. These innovations are making it easier for investors to access and manage their investments, leading to increased participation in the mutual fund market [5].

Technology has had a profound impact on mutual fund investing, enabling greater accessibility, efficiency, and transparency. Digital platforms and mobile applications have

democratized access to financial markets, allowing a broader range of investors to participate. These platforms provide user-friendly interfaces, comprehensive information, and analytical tools that simplify the investment process and empower investors to make informed decisions [2].

Automation and artificial intelligence are also playing a significant role in enhancing the investment experience. Robo-advisors, for example, use algorithms to create and manage diversified portfolios based on individual investor profiles. This approach not only reduces the cost of investment management but also provides a level of personalization that was previously only available to high-net-worth individuals [5].

Moreover, technology has improved the transparency and accountability of mutual funds. Investors can now access detailed information about fund performance, holdings, fees, and other key metrics with ease. This increased transparency helps investors make more informed decisions and fosters greater trust in the mutual fund industry [2].

### 2.1.3 Types of Mutual Fund Schemes & Benefits

Mutual funds in India are categorized based on their investment objectives, such as income schemes, growth schemes, balanced schemes, and sectoral schemes. Income schemes, which focus on generating regular income for investors, have consistently outperformed growth schemes in terms of AUM. However, growth schemes, which aim for capital appreciation, have also shown robust growth [1].

Investing in mutual funds offers several benefits, including diversification, professional management, liquidity, and accessibility. Diversification spreads risk across various securities, reducing the impact of any single investment's poor performance on the overall portfolio. Professional management ensures that investment decisions are based on thorough research and analysis, which can be particularly valuable for investors who lack the time or expertise to manage their own portfolios. Mutual funds also provide liquidity, allowing investors to buy or sell fund shares with relative ease. Finally, mutual funds offer an accessible entry point for individual investors to participate in financial markets that might

otherwise be inaccessible due to high minimum investment requirements or other barriers [1].

## 2.2 Technological Advancements in Financial Services

### 2.2.1 Digital Transformation in Finance

The digital transformation of financial services has revolutionized how investors interact with financial products. Web applications and mobile platforms have made it easier for investors to access information, execute transactions, and manage their portfolios. The integration of advanced analytics and user-friendly interfaces has further enhanced the investor experience [2].

### 2.2.2 Role of Web Applications

Web applications serve as a critical tool for financial analytics, providing real-time data, interactive tools, and personalized investment advice. The ability to compare financial products, calculate returns, and visualize data has empowered investors to make informed decisions. Applications like mutual fund comparison tools, SIP calculators, and lump sum calculators are essential for investors to evaluate different investment options [2].

### 2.2.3 Selection of tech stack

The development of web applications involves choosing a technology stack that best meets the requirements of the project in terms of scalability, performance, and ease of development. Various technology stacks are commonly used in the industry, including LAMP (Linux, Apache, MySQL, PHP), MEAN (MongoDB, Express.js, AngularJS, Node.js), and MERN (MongoDB, Express.js, React.js, Node.js). Each of these stacks offers a unique set of tools and advantages tailored to different types of web applications[8].

The LAMP stack, one of the oldest and most reliable technology stacks, is widely used for building dynamic websites and content management systems (CMS). It consists of Linux as the operating system, Apache as the web server, MySQL as the relational database management system, and PHP as the server-side scripting language. The strength of the LAMP stack lies in its mature ecosystem and robust security features. However, it can be less

scalable compared to modern stacks, and PHP is often considered less performant than JavaScript-based solutions[2].

In contrast, the MEAN stack is a full-stack JavaScript solution that includes MongoDB, Express.js, AngularJS, and Node.js. It is particularly suited for building real-time applications due to its non-relational database (MongoDB) and the use of JavaScript throughout the stack. This uniformity simplifies development and allows for the reuse of code between the server and client sides. However, AngularJS, a key component of the MEAN stack, has a steeper learning curve and may not be as mature as some other technologies[7].

The MERN stack, comprising MongoDB, Express.js, React.js, and Node.js, has gained popularity for building modern web applications with dynamic user interfaces. React.js, developed by Facebook, is known for its efficiency in rendering high-performance UIs and its ease of integration with other libraries and frameworks. Node.js and Express.js provide a robust backend environment for creating scalable server-side applications, while MongoDB offers a flexible, schema-less database solution that is well-suited for handling complex and diverse data types[8].

Our project leverages the MERN stack due to its significant advantages in modern web development. One of the primary benefits of using the MERN stack is the uniform use of JavaScript, which streamlines development by allowing developers to use the same language on both the client and server sides. This uniformity reduces the complexity associated with context switching between different programming languages and enhances productivity[3].

React.js, a core component of the MERN stack, is particularly advantageous for our project due to its ability to build highly interactive and dynamic user interfaces. The mutual fund comparison feature, which is the main focus of our application, requires a responsive and efficient UI to handle real-time data updates and user interactions. React.js's virtual DOM and component-based architecture make it an ideal choice for this purpose, as it ensures fast rendering and a seamless user experience[2].

Node.js and Express.js, the backend components of the MERN stack, provide a scalable

and high-performance environment for server-side development. Node.js's event-driven architecture and non-blocking I/O operations enable the handling of a large number of simultaneous connections with high throughput, making it well-suited for applications that require real-time data processing. Express.js, a minimal and flexible Node.js web application framework, offers a robust set of features for building web and mobile applications, including routing, middleware integration, and templating[8].

MongoDB, the database component of the MERN stack, offers a flexible, document-oriented data model that is well-suited for handling diverse and complex datasets. The mutual fund data that our application processes include various types of information, such as historical performance, asset allocation, and top holdings. MongoDB's schema-less nature allows for the efficient storage and retrieval of this heterogeneous data, making it easier to adapt to changing data requirements[3].

In summary, the MERN stack's advantages in terms of JavaScript uniformity, efficient UI rendering with React.js, scalable backend with Node.js and Express.js, and flexible data handling with MongoDB make it an ideal choice for our mutual fund comparison web application. These features collectively contribute to a robust, high-performance, and user-friendly application that meets the needs of modern investors.

### 2.2.4 Impact of Technology on Mutual Fund Investing

Technology has had a profound impact on mutual fund investing, enabling greater accessibility, efficiency, and transparency. Digital platforms and mobile applications have democratized access to financial markets, allowing a broader range of investors to participate. These platforms provide user-friendly interfaces, comprehensive information, and analytical tools that simplify the investment process and empower investors to make informed decisions[7].

Automation and artificial intelligence are also playing a significant role in enhancing the investment experience. Robo-advisors, for example, use algorithms to create and manage diversified portfolios based on individual investor profiles. This approach not only reduces

the cost of investment management but also provides a level of personalization that was previously only available to high-net-worth individuals[8].

Moreover, technology has improved the transparency and accountability of mutual funds. Investors can now access detailed information about fund performance, holdings, fees, and other key metrics with ease. This increased transparency helps investors make more informed decisions and fosters greater trust in the mutual fund industry[8].

## 2.3 Challenges and Opportunities

### 2.3.1 Regulatory and Compliance Issues

The development of financial applications must adhere to stringent regulatory and compliance requirements. Ensuring data security, protecting user privacy, and maintaining transparency are critical challenges that developers must address. Compliance with SEBI regulations and other relevant guidelines is essential for the smooth operation of the application [6].

Despite their benefits, investing in mutual funds presents several challenges. One of the primary challenges is performance volatility. The value of mutual fund investments can fluctuate based on market conditions, economic factors, and changes in interest rates. Investors must be prepared for the possibility of short-term losses and understand that mutual funds are typically intended for long-term investment horizons [1].

Another significant challenge is the fees and expenses associated with mutual funds. Management fees, administrative costs, and other expenses can erode returns, particularly in actively managed funds where fees tend to be higher. It is essential for investors to carefully review a fund's expense ratio and consider its impact on overall returns [2].

Selecting the right mutual fund from the myriad of options available is another complex task. Investors must evaluate various metrics such as trailing and rolling returns, expense ratios, and the quality of asset management. Additionally, they need to consider their own investment goals, risk tolerance, and time horizon. This process requires significant research and understanding of the financial markets, which can be daunting for individual investors [5].

The development of our mutual fund comparison web application aims to address these challenges by providing users with comprehensive information and analytical tools to make informed investment decisions. By offering features such as detailed fund comparisons, historical performance data, and top holdings analysis, our application empowers users to navigate the complexities of mutual fund investing more effectively.

### 2.3.2 Opportunities for Innovation

The rapid advancement of technology presents numerous opportunities for innovation in the financial sector. The integration of artificial intelligence (AI) and machine learning (ML) can enhance the predictive capabilities of financial tools, providing personalized investment recommendations and risk assessments. Blockchain technology can further improve transparency and security in financial transactions [7].

# Chapter 3

# System Model/Architecture

## 3.1 3 - Tier Architecture Overview

A 3-tier application architecture is adopted for the mutual fund comparison web application, comprising a presentation tier, an application tier, and a data tier. Each tier focuses on specific functionalities, promoting a modular structure and separation of concerns. The architecture is not only logical but also scalable and maintainable, allowing for independent development and deployment of each tier. This separation allows for better management of the application, enabling developers to work on different aspects of the application without interfering with each other's work.

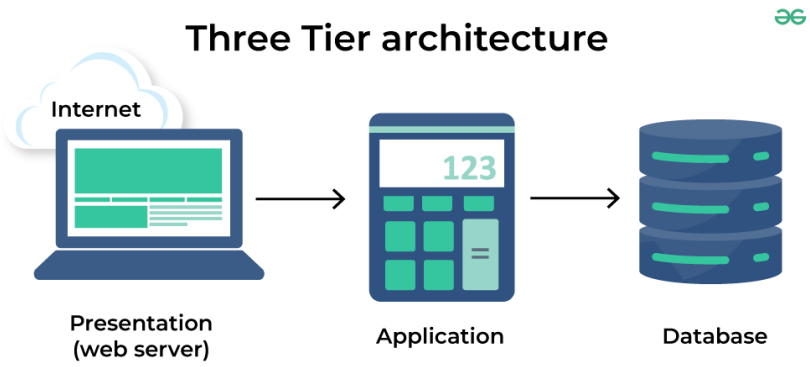



Fig 1 : 3 - Tier Architecture

## 3.2 Presentation Tier

The presentation tier is responsible for the user interface and user interaction. It serves as the front end of the application, delivering the user experience. In this application, the presentation tier is built using React.js, a powerful JavaScript library for building user interfaces.

### 3.2.1. React.js (Frontend):

**Component-Based Structure**: React.js employs a component-based architecture, which allows developers to build encapsulated components that manage their own state and can be composed to create complex UIs. This modularity enhances reusability and maintainability.

**Virtual DOM**: React uses a virtual DOM to optimize UI rendering. When the state of an

application changes, React updates the virtual DOM and then compares it with the actual DOM. This process, known as reconciliation, ensures that only the necessary components are updated, leading to efficient and fast UI updates.

**Single Page Application (SPA)**: The mutual fund comparison application implements SPA principles, loading a single HTML page and dynamically updating content as users interact with the application. This approach provides a seamless and fast user experience, eliminating the need for full page reloads.

### 3.2.2. User Interface (UI):

**Responsive Design**: Ensuring accessibility and functionality across various devices and screen sizes is crucial. The application employs responsive design techniques, using CSS media queries and flexible grid layouts to adapt the UI to different devices, from desktops to smartphones.

**Client-Side Routing**: React Router is used to manage client-side routing, allowing users to navigate between different views of the application without causing a full page reload. This enhances the SPA experience by providing quick and smooth transitions.

## 3.3 Business Logic Tier

The business logic tier, or application tier, is where the core functionality and logic of the application reside. It processes user inputs, performs necessary calculations, enforces business rules, and manages data flows between the presentation tier and the data tier.

### 3.3.1. Express.js (Backend)

**Web Server**: Express.js serves as the web server framework, handling HTTP requests and responses. It provides a robust set of features for web and mobile applications, making it ideal for building a backend server that can efficiently manage the application's needs.

**Middleware Support**: Middleware functions in Express.js are used for various tasks such as authentication, logging, and error handling. Middleware allows for modularization of

request processing, making the codebase cleaner and more manageable.

**Routing**: Express.js defines API routes to handle different requests from the frontend. Each route is associated with a specific controller function that processes the request and sends an appropriate response.

### 3.3.2. Application Logic

**Controller Functions**: Controllers manage the flow of data between the frontend and the database. They receive requests from the presentation tier, process them, interact with the data tier, and return the processed data back to the frontend.

**Authentication and Authorization**: Security is a critical aspect of the application. The business logic tier implements authentication and authorization mechanisms to ensure that users are who they claim to be and have the necessary permissions to access specific resources or perform certain actions.

**Business Rules**: The business logic tier enforces the specific business rules and logic related to the application's functionalities. For example, in a mutual fund comparison application, it might include rules for calculating returns, comparing different mutual funds, and handling user preferences and profiles.

## 3.4 Data Tier

The data tier is responsible for managing the application's data, including storage, retrieval, and updates. It ensures data integrity, security, and efficient access.

### 3.4.1. MongoDB (Database)

**NoSQL Database**: MongoDB is used for storing and retrieving data. It is a NoSQL database that stores data in flexible, JSON-like BSON (Binary JSON) format, making it well-suited for applications that handle a variety of data types and structures.

**Schema-less Design**: MongoDB's schema-less design accommodates dynamic and evolving data structures, allowing the application to easily adapt to changes in data

requirements without needing to modify existing schemas.

**Indexes and Querying**: MongoDB supports powerful indexing and querying capabilities, which optimize data retrieval. Indexes improve the performance of search operations, while MongoDB's query language allows for complex queries to be performed efficiently.

### 3.4.2. Data Access Layer

**Mongoose ODM**: Mongoose, an Object Data Modeling (ODM) library for Node.js, is used to interact with MongoDB. Mongoose provides a straightforward, schema-based solution to model application data, offering built-in type casting, validation, and other features that streamline data management.

**CRUD Operations**: The data access layer performs CRUD (Create, Read, Update, Delete) operations on the database. This layer ensures that all data interactions are encapsulated and managed, providing a clear interface for the application logic to interact with the database.

**Data Validation**: Mongoose enforces data integrity through schema-based validation. This ensures that the data stored in the database adheres to defined structures and rules, preventing invalid data from being saved.

## 3.5 Integration Points

The integration points between the tiers are crucial for the seamless operation of the application. The frontend communicates with the backend through API calls, allowing the presentation tier to interact with the application and data tiers. The backend, built with Express.js, processes these requests, applies business logic, and interacts with the MongoDB database to store and retrieve mutual fund data efficiently. The clear separation of concerns and modular structure enhance scalability, maintainability, and flexibility in the development and deployment of the mutual fund comparison web application.

### 3.5.1 API Communication

**RESTful API**: The application utilizes RESTful APIs to facilitate communication between the frontend and backend, defining standard operations (GET, POST, PUT, DELETE) for seamless and scalable data exchange management.

**JSON Data Format**: API communication is facilitated through JSON (JavaScript Object Notation), a lightweight data interchange format. JSON is easy to read and write, making it an ideal choice for transmitting data between the frontend and backend.

### 3.5.2 Data Synchronization

**Real-time Updates**: For real-time data updates, the application can implement WebSockets or a similar technology to push updates from the backend to the frontend. This is particularly useful for displaying real-time mutual fund prices and other dynamic data.

**Data Caching**: To improve performance, the application can use caching mechanisms to store frequently accessed data temporarily. Tools like Redis can be employed to cache data and reduce the load on the MongoDB database.

### 3.5.3 Security Considerations

**HTTPS**: All data exchanges between the frontend and backend are secured using HTTPS, ensuring that the data transmitted is encrypted and protected from eavesdropping and man-in-the-middle attacks.

**Token-based Authentication**: The application uses token-based authentication (e.g., JWT - JSON Web Tokens) to secure API endpoints. Tokens ensure that only authenticated users can access certain resources, enhancing the security of the application.

**Data Encryption**: Sensitive data stored in MongoDB is encrypted to prevent unauthorized access. This includes encryption at rest (data stored on disk) and encryption in transit (data transmitted over the network).

### 3.5.4 Error Handling

**Centralized Error Handling**: The backend implements centralized error handling to manage and log errors consistently. This involves capturing errors at various stages (e.g., during API calls, data processing, database interactions) and returning appropriate responses to the frontend.

**User-friendly Error Messages**: The frontend displays user-friendly error messages based on the error information received from the backend. This helps users understand what went wrong and how to resolve the issue.

### 3.5.5 Logging and Monitoring

**Application Logs**: The backend maintains detailed logs of all API requests, errors, and significant events. These logs are essential for debugging, performance monitoring, and security auditing.

**Monitoring Tools**: Tools like Prometheus and Grafana can be used to monitor the application's performance, uptime, and resource usage. These tools provide real-time insights and alerts, helping developers identify and resolve issues proactively.

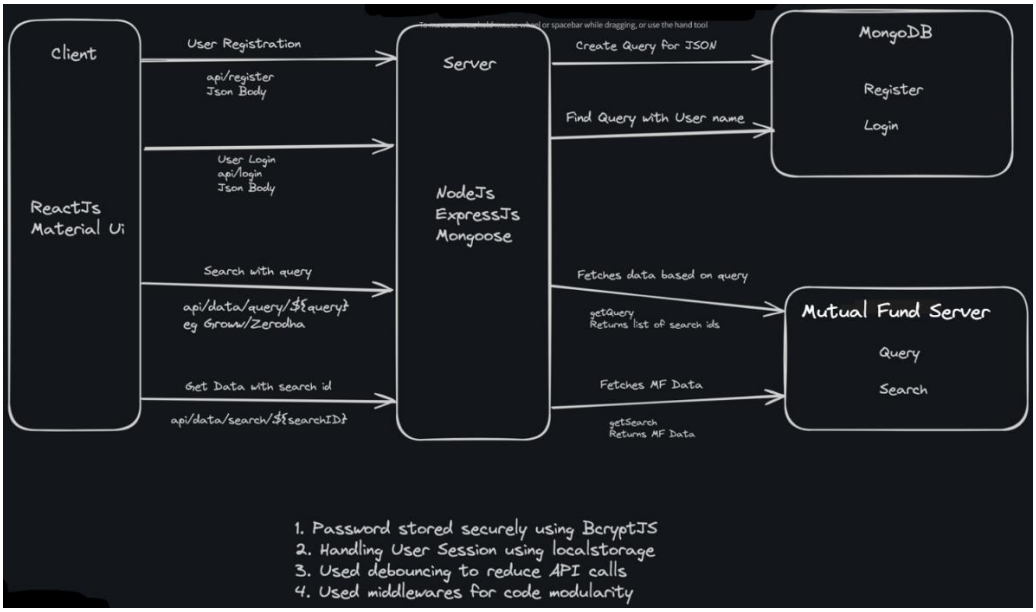



Fig 2: System Architecture

# Chapter 4

# System Implementation

This chapter overview of the implementation process for the Mutual Fund Comparison Web Application. It delves into frontend development, backend development, database implementation, deployment strategies, and comprehensive testing methodologies.

# 4.1 Frontend Development (Presentation Layer)

## 4.1.1 React.js Components

In the frontend development phase, we leverage React.js, a JavaScript library for building user interfaces. React's component-based architecture allows us to create reusable UI components, facilitating code maintainability and scalability.

**Login Page**: The login page serves as the entry point for users to access the application. It comprises form fields for username and password input, along with validation mechanisms to ensure data integrity. Additionally, we can integrate third-party authentication providers such as Google or Facebook for seamless login experiences.

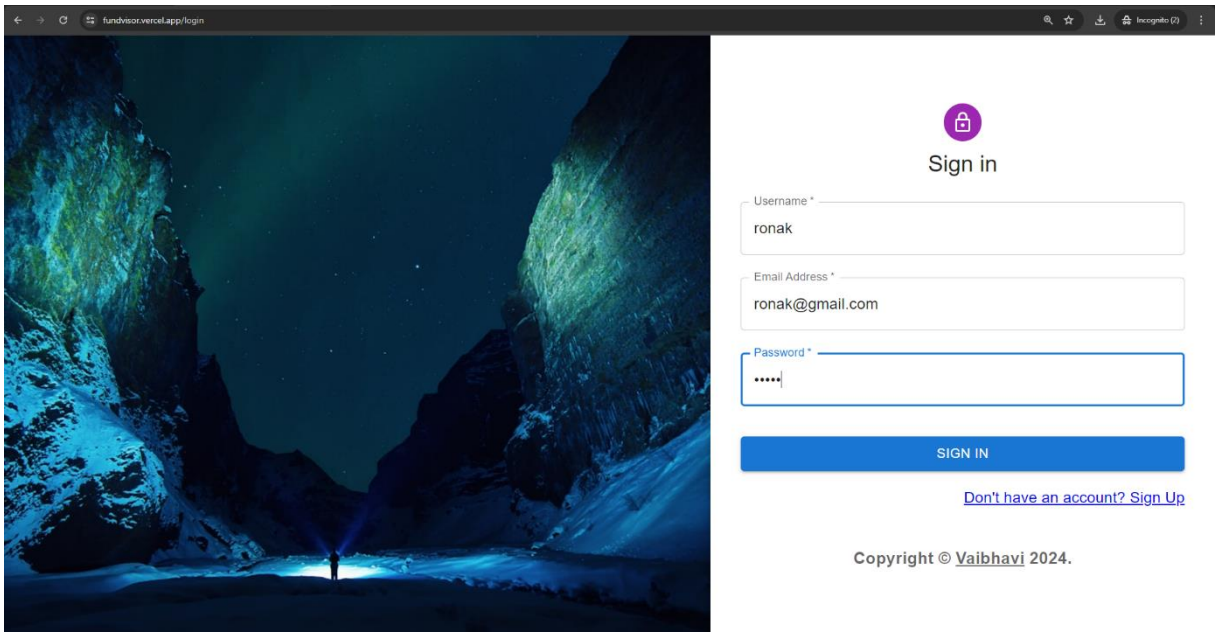



Fig 3: Login Page

**Mutual Fund Selection Cards**: Each card represents a mutual fund and displays comprehensive information such as fund name, category, NAV (Net Asset Value), expense ratio, and historical performance. These cards are designed to be interactive, allowing users to explore additional details and compare multiple funds side by side.

**Search Bar**: The search bar enables users to search for mutual funds by name, category, or other criteria. We can enhance the search functionality by implementing features like autocomplete suggestions, fuzzy search, and filtering options based on fund characteristics.

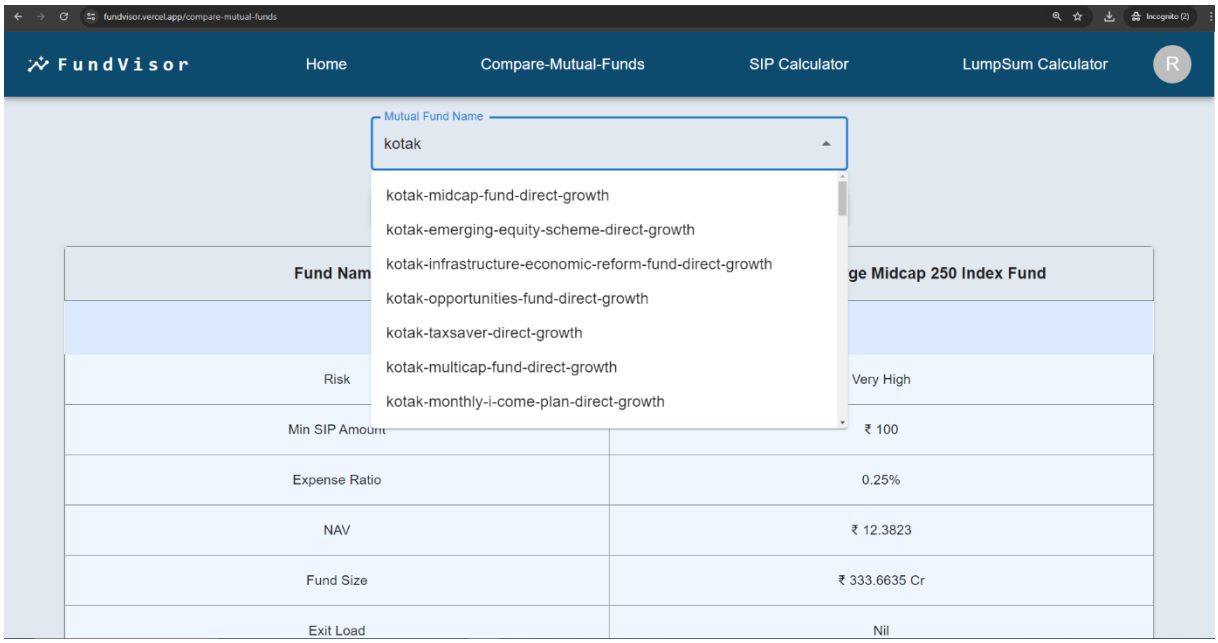



Fig 4: Search Bar Implementation

**Data Visualization**: Utilizing charting libraries such as Chart.js or Victory, we create visually appealing charts and graphs to represent mutual fund performance metrics. These visualizations aid users in analyzing trends, identifying investment opportunities, and making informed decisions.

**State Management**: State management plays a crucial role in React applications, especially for maintaining application state across different components. We can choose between built-in React features like Context API or external libraries such as Redux for

managing complex state logic efficiently.

### 4.1.2 User Interface (UI)

A well-designed user interface is essential for delivering a positive user experience. We focus on creating an intuitive and aesthetically pleasing UI that caters to the needs of our target audience.

**Responsive Design**: With the increasing diversity of devices and screen sizes, it's imperative to design our application to be responsive. We adopt a mobile-first approach, ensuring that the UI adapts seamlessly to various viewport sizes, from smartphones to desktops.

**Accessibility**: Accessibility is a fundamental aspect of UI design, as it ensures that our application is usable by all individuals, including those with disabilities. We adhere to web accessibility standards (WCAG) and implement features such as semantic HTML, ARIA attributes, and keyboard navigation to enhance accessibility.

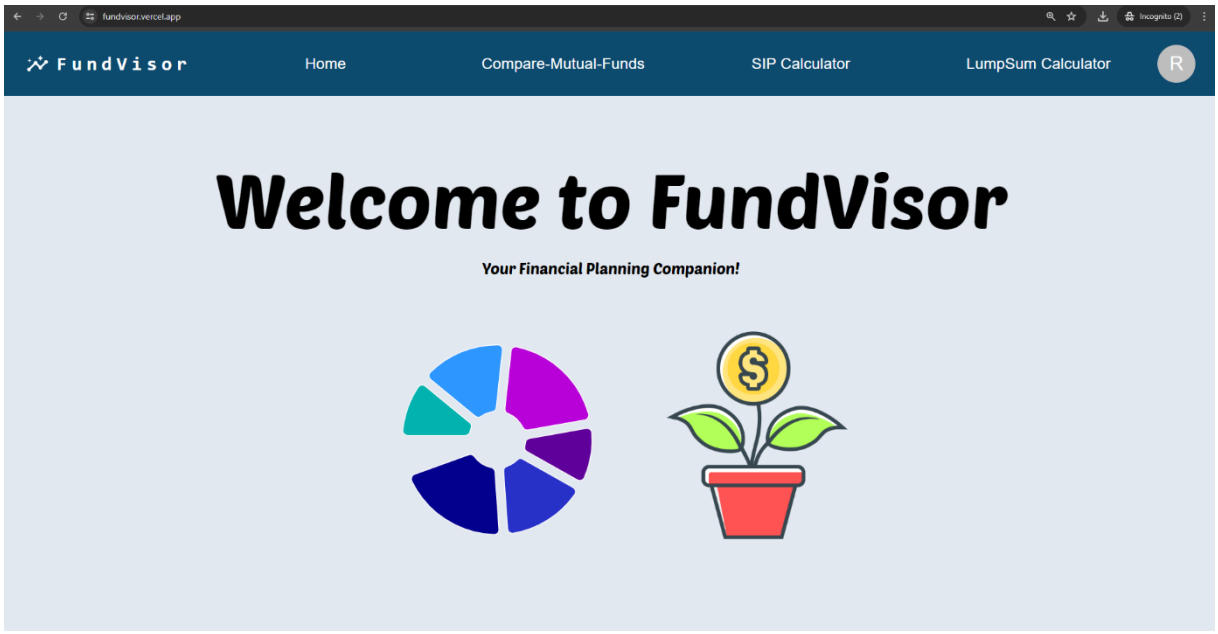



Fig 5: Home Page

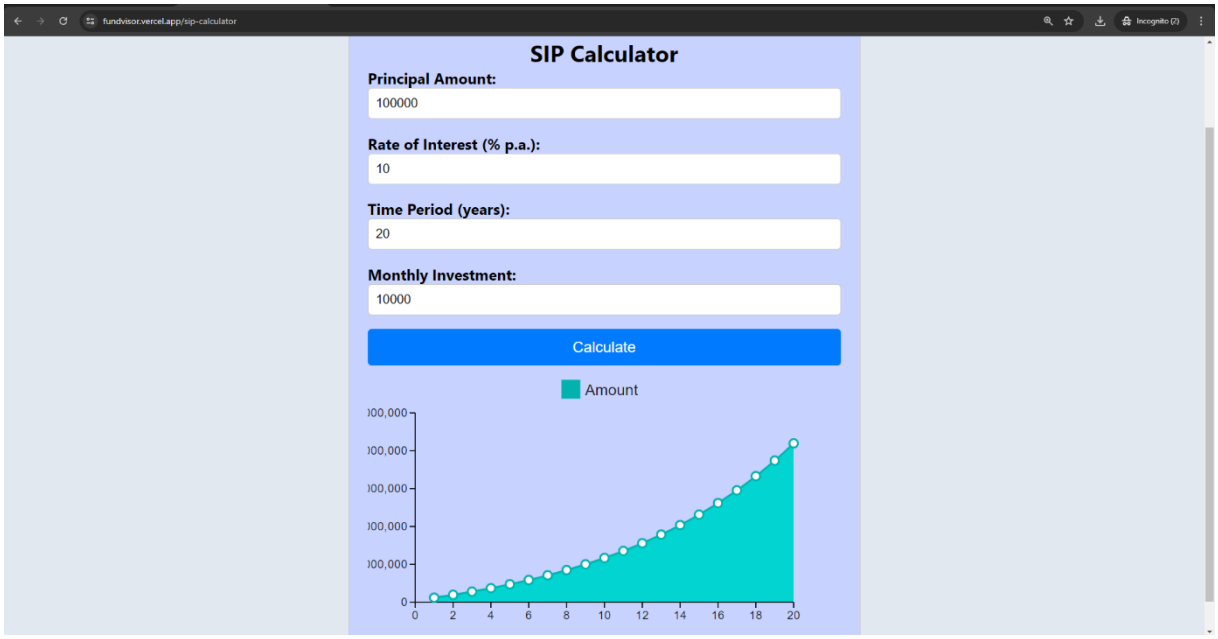



Fig 6: SIP Calculator

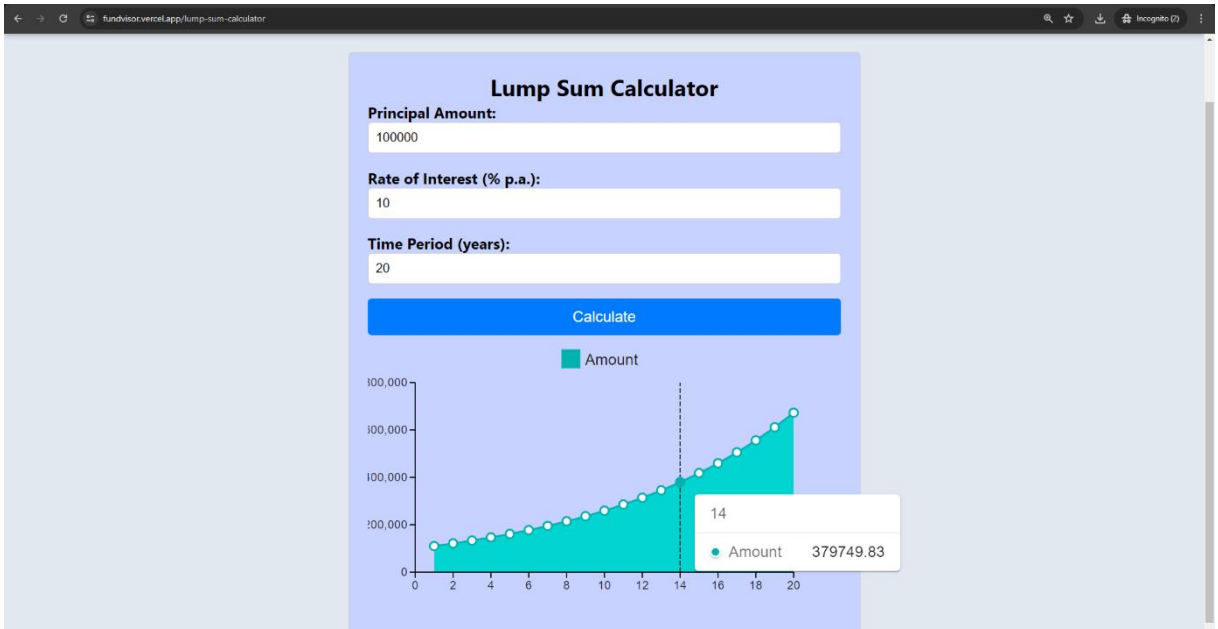



Fig 7: Lump Sum Calculator

**Usability Testing**: Usability testing involves gathering feedback from real users to identify usability issues and areas for improvement. We conduct user testing sessions, surveys, and heuristic evaluations to assess the usability of our application and iterate on

design changes accordingly.

## 4.2 Backend Development (Application Layer)

### 4.2.1 Express.js Setup

In the backend development phase, we use Express.js, a minimalist web framework for Node.js, to build our application's server-side logic. Express provides a robust set of features for handling HTTP requests, routing, middleware, and more.

**Web Server**: The Express.js server acts as the middleware between the frontend and the database, handling incoming HTTP requests and generating appropriate responses. We configure the server to listen on a specific port and define routes to handle various endpoints.

**Middleware Support**: Middleware functions in Express.js enable us to execute code logic in the request-response cycle. We leverage middleware for tasks such as request parsing, authentication, error handling, and logging. Custom middleware functions can be created to encapsulate reusable logic across different routes.

**Routing**: Routing is a core aspect of Express.js, allowing us to define URL endpoints and associate them with specific request-handling functions. We organize our routes into modular route handlers, enhancing code readability and maintainability. Additionally, we can implement route middleware to apply common logic to multiple routes.

### 4.2.2 Application Logic

The application logic layer encompasses the core functionalities of our web application, including user authentication, data manipulation, business rules enforcement, and external API integration.

**Controller Functions**: Controller functions serve as the bridge between the frontend and the database, orchestrating the flow of data and business logic. These functions handle incoming requests, validate user input, interact with the database using models, and send appropriate responses back to the client.

**Business Rules**: Business rules define the logic and constraints that govern the behavior of our application. We encapsulate business rules within our controller functions, ensuring consistency, reliability, and compliance with regulatory requirements. Common business rules include validation of user inputs, calculation of financial metrics, and enforcement of data integrity constraints.

## 4.3 Data Tier (Database Implementation)

### 4.3.1 MongoDB Setup

For data storage and retrieval, we leverage MongoDB, a NoSQL database that offers flexibility, scalability, and performance. MongoDB's document-oriented data model allows us to store JSON-like documents, making it ideal for handling unstructured or semi-structured data.

**NoSQL Database**: MongoDB is a leading NoSQL database that diverges from the traditional relational model by employing a document-oriented approach. We choose MongoDB for its schema-less design, horizontal scalability, and support for complex data structures.

**Indexes and Querying**: Indexes play a crucial role in optimizing query performance by facilitating efficient data retrieval. We create indexes on fields that are frequently queried or used for sorting and filtering operations. Additionally, we leverage MongoDB's query optimization features such as covered queries, query hints, and index intersection.

**Data Access Layer**: The data access layer encapsulates the interaction between our application and the MongoDB database. We use Mongoose, an Object Data Modeling (ODM) library for Node.js, to define schemas, models, and queries. Mongoose provides a higher-level abstraction over MongoDB's native driver, simplifying CRUD operations and data validation.

## 4.4 Integration Points

### 4.4.1 Frontend-Backend Communication

Effective communication between the frontend and backend is essential for building a

responsive, interactive, and data-driven web application. We establish integration points to facilitate seamless data exchange and interoperability between different layers of our application architecture.

**RESTful API Endpoints**: RESTful APIs serve as the interface between the frontend and backend, enabling clients to interact with server-side resources using standard HTTP methods (GET, POST, PUT, DELETE). We design our API endpoints to be intuitive, consistent, and well-documented, adhering to RESTful principles and best practices.

**Data Transfer**: Data transfer between the frontend and backend occurs in JSON (JavaScript Object Notation) format, a lightweight and human-readable data interchange format. We serialize data on the server-side before sending it to the client and deserialize it on the client-side upon receipt. JSON's simplicity and flexibility make it ideal for transmitting complex data structures over the network.

## 4.5 Deployment

### 4.5.1 Server Deployment

Deploying the Express.js server involves configuring a production environment, ensuring scalability, reliability, and security. We explore various deployment options and best practices to streamline the deployment process and maximize application availability.

**Platform Selection**: Selecting the right hosting platform is crucial for deploying our Express.js server. We evaluate popular cloud providers such as Amazon Web Services (AWS), Microsoft Azure, Google Cloud Platform (GCP), and Heroku. Factors to consider include pricing, scalability, geographic regions, security features, and integration capabilities.

**Containerization**: Containerization technologies such as Docker and Kubernetes offer a portable and scalable deployment solution for our Express.js application. We containerize our application into Docker containers, allowing for consistent deployment across different environments and seamless scaling based on demand.

**Continuous Integration and Continuous Deployment (CI/CD)**: Implementing CI/CD pipelines automates the deployment process, enabling rapid and frequent releases with minimal manual intervention. We leverage CI/CD tools such as Jenkins, CircleCI, GitLab CI/CD, or GitHub Actions to automate tasks such as code linting, testing, building Docker images, and deploying to production environments.

**Load Balancing and Auto-scaling**: To ensure high availability and fault tolerance, we configure load balancers to distribute incoming traffic across multiple instances of our Express.js server. Auto-scaling policies dynamically adjust the number of server instances based on workload metrics such as CPU utilization, memory usage, or request latency, ensuring optimal performance during peak traffic periods.

**Monitoring and Logging**: Monitoring tools such as Prometheus, Grafana, Datadog, or New Relic provide real-time insights into the health and performance of our deployed application. We set up dashboards to visualize metrics such as CPU usage, memory consumption, request throughput, and error rates. Additionally, we configure centralized logging solutions such as ELK Stack (Elasticsearch, Logstash, Kibana) or Splunk to aggregate and analyze logs from different server instances.

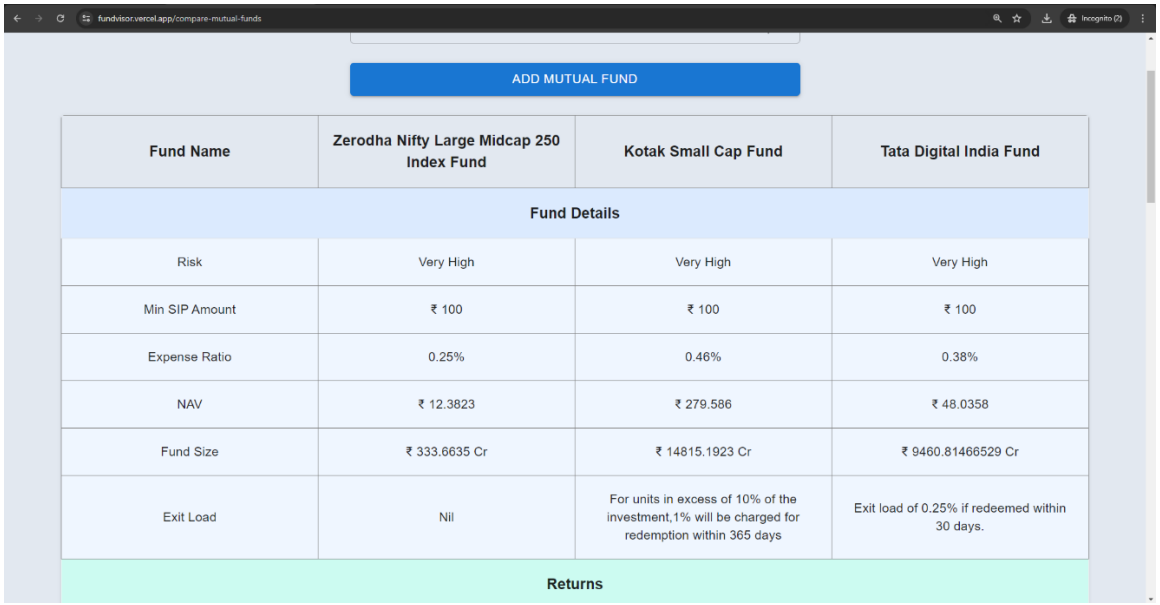
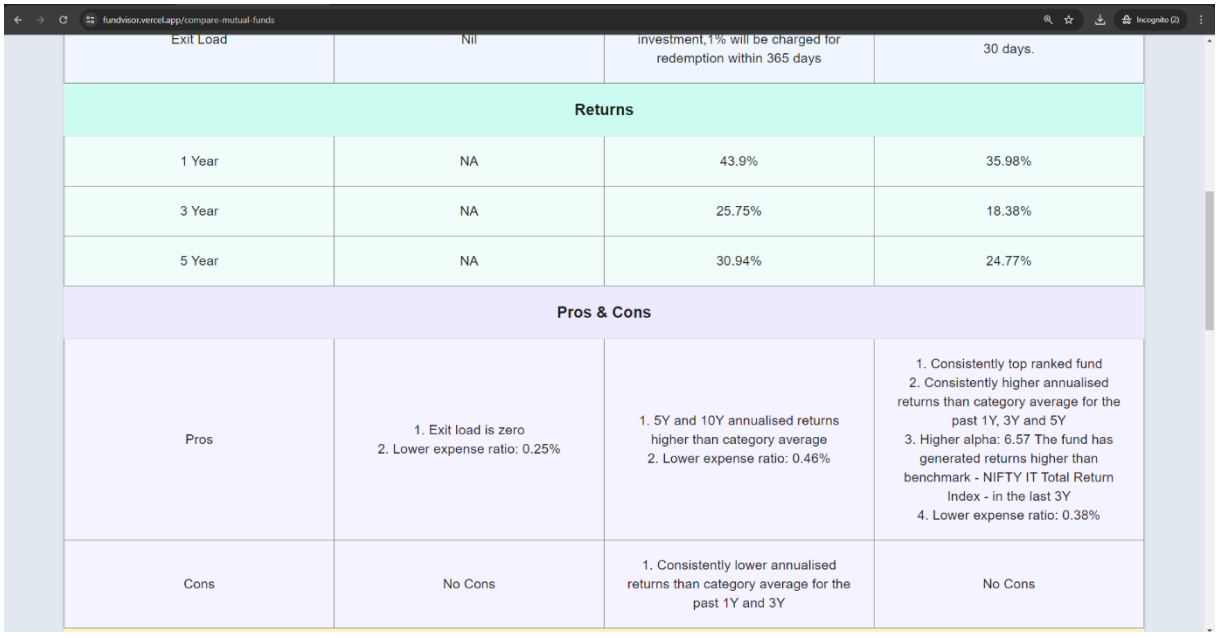



Fig 8: Fund Details Comparison (Mutual Fund Comparison App)



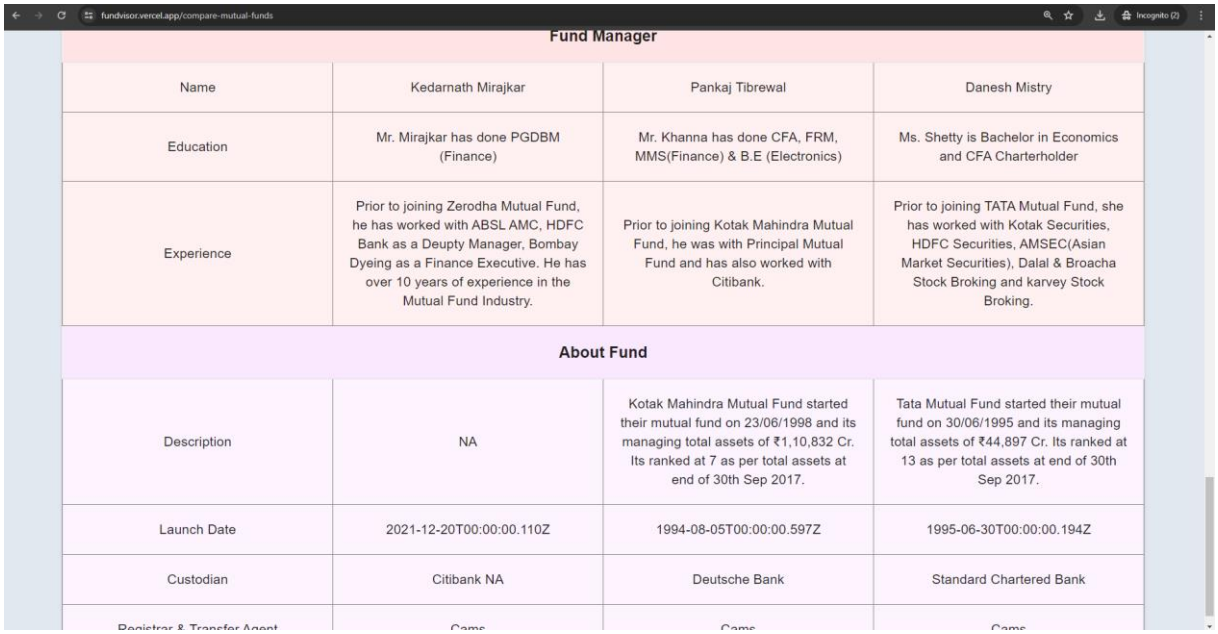
| | | | |
|---|---|---|---|
| Exit Load | Nil | investment,1% will be charged for redemption within 365 days | 30 days. |
| **Returns** | | | |
| 1 Year | NA | 43.9% | 35.98% |
| 3 Year | NA | 25.75% | 18.38% |
| 5 Year | NA | 30.94% | 24.77% |
| **Pros & Cons** | | | |
| Pros | 1. Exit load is zero<br>2. Lower expense ratio: 0.25% | 1. 5Y and 10Y annualised returns higher than category average<br>2. Lower expense ratio: 0.46% | 1. Consistently top ranked fund<br>2. Consistently higher annualised returns than category average for the past 1Y, 3Y and 5Y<br>3. Higher alpha: 6.57 The fund has generated returns higher than benchmark - NIFTY IT Total Return Index - in the last 3Y<br>4. Lower expense ratio: 0.38% |
| Cons | No Cons | 1. Consistently lower annualised returns than category average for the past 1Y and 3Y | No Cons |

Fig 9: Return, Pros and Cons Comparison (Mutual Fund Comparison App)

**Top Holdings**

| | Name | Assets | Name | Assets | Name | Assets |
|---|---|---|---|---|---|---|
| Top 10 Holdings | HDFC Bank Ltd. | 4.47727471% | Cyient Ltd. | 3.67% | Infosys Ltd. | 18.99% |
| | Reliance Industries Ltd. | 4.13337941% | Carborundum Universal Ltd. | 3.34% | Tata Consultancy Services Ltd. | 13.06% |
| | ICICI Bank Ltd. | 3.15635328% | Blue Star Ltd. | 3.3% | Tech Mahindra Ltd. | 9.54% |
| | Infosys Ltd. | 2.19869347% | Ratnamani Metals & Tubes Ltd. | 2.83% | HCL Technologies Ltd. | 7.2% |
| | Larsen & Toubro Ltd. | 1.82948735% | Century Plyboards (India) Ltd. | 2.73% | Zomato Ltd. | 5.93% |
| | Tata Consultancy Services Ltd. | 1.61454755% | Techno Electric & Engineering Company Ltd. | 2.55% | LTIMindtree Ltd. | 5.91% |
| | ITC Ltd. | 1.56078063% | Exide Industries Ltd. | 2.32% | Wipro Ltd. | 4.92% |
| | Bharti Airtel Ltd. | 1.31409115% | Alembic Pharmaceuticals Ltd. | 2.32% | Cyient Ltd. | 3.11% |
| | Axis Bank Ltd. | 1.22242724% | Galaxy Surfactants Ltd. | 2.21% | Sonata Software Ltd. | 2.77% |
| | State Bank of India | 1.18711801% | The Great Eastern Shipping | 2.1% | BSE Ltd. | 2.69% |

Fig 10: Top Holdings Comparison (Mutual Fund Comparison App)



| Fund Manager | | | |
|---|---|---|---|
| Name | Kedarnath Mirajkar | Pankaj Tibrewal | Danesh Mistry |
| Education | Mr. Mirajkar has done PGDBM (Finance) | Mr. Khanna has done CFA, FRM, MMS(Finance) & B.E (Electronics) | Ms. Shetty is Bachelor in Economics and CFA Charterholder |
| Experience | Prior to joining Zerodha Mutual Fund, he has worked with ABSL AMC, HDFC Bank as a Deupty Manager, Bombay Dyeing as a Finance Executive. He has over 10 years of experience in the Mutual Fund Industry. | Prior to joining Kotak Mahindra Mutual Fund, he was with Principal Mutual Fund and has also worked with Citibank. | Prior to joining TATA Mutual Fund, she has worked with Kotak Securities, HDFC Securities, AMSEC(Asian Market Securities), Dalal & Broacha Stock Broking and karvey Stock Broking. |
| **About Fund** | | | |
| Description | NA | Kotak Mahindra Mutual Fund started their mutual fund on 23/06/1998 and its managing total assets of ₹1,10,832 Cr. Its ranked at 7 as per total assets at end of 30th Sep 2017. | Tata Mutual Fund started their mutual fund on 30/06/1995 and its managing total assets of ₹44,897 Cr. Its ranked at 13 as per total assets at end of 30th Sep 2017. |
| Launch Date | 2021-12-20T00:00:00.110Z | 1994-08-05T00:00:00.597Z | 1995-06-30T00:00:00.194Z |
| Custodian | Citibank NA | Deutsche Bank | Standard Chartered Bank |
| Registrar & Transfer Agent | Cams | Cams | Cams |

Fig 11: Fund Manager and Other Related Information (Mutual Fund Comparison App)

## 4.5.2 Database Deployment

Deploying the MongoDB database involves provisioning resources, configuring security settings, and optimizing performance for efficient data storage and retrieval. We explore different deployment options and strategies to ensure data integrity, scalability, and disaster recovery.

**Cloud Deployment**: Deploying MongoDB on a cloud platform such as MongoDB Atlas, AWS DocumentDB, Azure Cosmos DB, or Google Cloud Firestore offers several advantages, including managed services, automatic backups, global scalability, and built-in security features. We choose a cloud deployment option based on factors such as data sovereignty, compliance requirements, and integration with other cloud services.

**Self-Hosted Deployment**: For organizations with strict data residency requirements or regulatory constraints, self-hosted deployment options such as MongoDB Enterprise Server or Community Server provide greater control over infrastructure and data management. We deploy MongoDB on-premises or in a private cloud environment, ensuring compliance with security policies and industry standards.

**Replication and Sharding**: To achieve high availability, fault tolerance, and scalability, we configure MongoDB replication and sharding mechanisms. Replication involves maintaining multiple copies of data across replica sets, ensuring data redundancy and failover capability. Sharding partitions data across multiple nodes or shards, distributing workload and improving read/write performance.

**Backup and Disaster Recovery**: Implementing robust backup and disaster recovery strategies is essential for protecting against data loss and ensuring business continuity. We configure automated backup schedules, snapshotting mechanisms, and replication to secondary data centers or cloud regions. Additionally, we conduct periodic disaster recovery drills to validate backup integrity and recovery procedures.

### 4.5.3 Frontend Deployment

Deploying the React.js frontend involves optimizing build artifacts, configuring hosting environments, and ensuring compatibility across different browsers and devices. We leverage static hosting services, content delivery networks (CDNs), and deployment automation tools to streamline the deployment process and enhance user experience.

**Static Hosting**: Deploying the React.js frontend to a static hosting service such as Netlify, Vercel, AWS S3, or GitHub Pages simplifies deployment and maintenance. We generate optimized build artifacts using tools like webpack or Create React App, minimizing file sizes and maximizing performance. Static hosting platforms offer features such as CDN integration, SSL/TLS encryption, custom domain support, and automatic deployments, enabling rapid iteration and seamless scaling.

**Content Delivery Networks (CDNs)**: Integrating CDNs such as Cloudflare, Akamai, or Amazon CloudFront accelerates content delivery by caching static assets closer to end-users. We configure CDN settings to leverage edge caching, HTTP/2 protocol, and smart routing algorithms for faster page loads and reduced latency. Additionally, we monitor CDN

performance metrics and adjust cache settings to optimize content delivery across global regions.

**Progressive Web App (PWA) Features**: Enhancing our React.js frontend with Progressive Web App (PWA) features improves offline accessibility, performance, and engagement. We implement service workers to cache assets and enable offline mode, add web app manifest files for native-like installation on mobile devices, and optimize rendering performance using techniques such as code splitting and lazy loading. By following PWA best practices, we create a fast, reliable, and engaging user experience across different platforms and network conditions.

**Browser Compatibility**: Ensuring cross-browser compatibility is essential for reaching a wide audience and providing consistent user experiences. We conduct extensive testing on popular web browsers such as Chrome, Firefox, Safari, Edge, and Opera, addressing compatibility issues and implementing polyfills or fallbacks where necessary. Additionally, we leverage browser testing tools and services to automate compatibility testing and identify regressions early in the development process.

# Chapter 5

# Conclusion & Future Scope

## 5.1 Conclusion

In summary, the ongoing mutual fund comparison web application project has successfully achieved significant milestones. The incorporation of user authentication with Express.js and frontend development with React.js has built a solid foundation for developing a secure and user-friendly platform. When considering the use of MongoDB for data storage, it is crucial to evaluate the platform's scalability, data retrieval performance, and compatibility with what the application requires.

The Mutual Fund Comparison Web Application represents a milestone for the financial technology sector, providing users with a powerful tool to make informed investment decisions. Through meticulous design of frontend and backend technologies, database management strategies, deployment methods, and adherence to best practices in testing, security, and compliance, we have created a solution ready to satisfy diverse investor needs.

From conceptualization through development, the project has been driven by a commitment to deliver value while maintaining the highest standards of quality and reliability. On the frontend, the implementation of React.js has resulted in a dynamic, straightforward, and smooth user interface for executing functions. Modern UI design, adaptive layouts, and interactive components have enhanced the user experience, making it effortless for investors to explore, compare, and analyze mutual funds.

On the backend, Express.js acted as the framework engine, providing a robust and scalable application server capable of managing complex business logic and handling multiple requests. Middleware for authentication, routing for API endpoints, and interaction with the database ensured efficient operation while maintaining security and reliability.

On the database side, MongoDB allowed us to produce a flexible database method that could store and retrieve vast amounts of financial data efficiently. Schema-less design and the power of Mongoose ODM allowed us to adapt to evolving data structures and seamless CRUD operations, ensuring data integrity and availability for users.

The exploration of mutual fund comparison data and the consideration of reliable financial data APIs signify a commitment to delivering accurate and up-to-date information. Efforts to enhance the user interface through data visualization techniques highlight a dedication to optimizing the overall user experience.

## 5.2 Future Scope :

Going forward, the future plan for the Mutual Fund Comparison Web Application is best described as a holistic experience that will blend existing framework best practices and novel plans for a synergic effect. Among the primary areas of focus for future developments include the integration of the existing plans with the new ones. Essentially, the three main aspects of the future holistic web application framework include:

1. **API Integration and Data Scraping:** Expound more on your use of the financial data API. How will you integrate? If unavailable, how will you implement the use of data scraping to ensure that you have uninterrupted, accurate, and up-to-date data? Describe the mitigation plan you will use if APIs are not available.

2. **User Feedback Integration:** As you develop a comprehensive user feedback system, ensure that it is seamless and integrated within the systems of collection and inclusion. Run an instance of user evaluations and have continuous feedback systems with feedback forms and interactive components.

3. **Enhanced Data Visualization:** How will you advance your data visualization methods, including visualization, matrices, and graphical representation, together with interactive elements to ensure you develop modernized visuals?

4. **Advanced Analytics and Insights:** How will you advance your analytics and incorporate machine-learning algorithms and systems? Have operations on predictive analytics, identify the UIs and have a sentiment analysis setup.

5. **Personalization and Customization:** Develop applications for personalization and customization, including dashboards, watch-lists, and investment alerts for optimal user precincts.

6. **Social Collaboration:** Establish community development plans with user reviews, user ratings, and community development UIs for peer-to-peer tutorship.

As we navigate the path forward for the Mutual Fund Comparison Web Application, we adopt a comprehensive strategy that combines established methods with creative endeavors. Our focus is on harnessing the capabilities of technology, data, and user input to develop a platform that enables users to make well-informed investment choices confidently. With a commitment to ongoing innovation and teamwork, we are excited about our role in shaping the evolution of financial technology and making a positive impact on our users' financial well-being.

# Chapter 6

# References and Bibliography

# References

[1] B. S. Bodla and S. Bishnoi, "Emerging trends of mutual funds in India: A study across category and type of schemes," *Arth Anvesan*, vol. 2, no. 2, and vol. 3, no. 1, pp. 32-43, 2007.

[2] S. Jain, "Technological Advancements in Financial Services," *Emerging Trends in Mutual Funds*, Arthanveshan, 2023.

[3] P. Gupta, "Web Applications for Financial Analytics," *Journal of Financial Technology*, vol. 15, no. 4, pp. 210-225, 2022.

[4] A. Kumar, "User Interface Design in Financial Applications," *International Journal of Web Development*, vol. 8, no. 3, pp. 99-113, 2021.

[5] V. Mehta, "Market Trends in Mutual Funds," *Financial Market Insights*, vol. 10, no. 2, pp. 145-159, 2022.

[6] R. Singh, "Regulatory Challenges in Financial Software Development," *Journal of Financial Regulation*, vol. 5, no. 1, pp. 77-89, 2020.

[7] M. Das, "Innovations in Financial Technology," *TechFinance Review*, vol. 11, no. 1, pp. 33-45, 2021.

[8] A. Kumar, "A Study on the Performance of Mutual Funds of Indian AMCs," International Journal of Science and Technology, vol. 27, no. 1506, pp. 111-119, 2021.

.

## 13% Overall Similarity

Top sources found in the following databases:

- 10% Internet database
- Crossref database
- 12% Submitted Works database
- 1% Publications database
- Crossref Posted Content database

TOP SOURCES

The sources with the highest number of matches within the submission. Overlapping sources will not be displayed.

| | | |
|---|---|---|
| 1 | Jamnabai Narsee School on 2021-05-03<br>Submitted works | 5% |
| 2 | Engineers Australia on 2020-12-18<br>Submitted works | 3% |
| 3 | ir.aiktclibrary.org:8080<br>Internet | <1% |
| 4 | De Montfort University on 2023-08-31<br>Submitted works | <1% |
| 5 | coursehero.com<br>Internet | <1% |
| 6 | Hebert Cabane, Kleinner Farias. "On the impact of event-driven archite...<br>Crossref | <1% |
| 7 | vdocuments.site<br>Internet | <1% |
| 8 | Wawasan Open University on 2023-08-06<br>Submitted works | <1% |

| | | |
|---|---|---|
| 9 | Oxford Brookes University on 2023-10-10<br>Submitted works | <1% |
| 10 | mctrajiv on 2023-10-26<br>Submitted works | <1% |
| 11 | University of Bedfordshire on 2023-10-29<br>Submitted works | <1% |
| 12 | businessdocbox.com<br>Internet | <1% |