# API DOCUMENTATION

# Dashboard API Documentation

Overview:  
The Dashboard API provides various functionalities to manage and retrieve data related to the user's investments, including performance indicators, fund management, stock information, and portfolio analysis.

## Endpoints:

### 1. GET /PerformanceIndicators

Description: Retrieves the total investment, current valuation, and calculates growth in absolute and percentage terms.  
Response:  
{  
 "moneyIn": 42000,  
 "valuation": 45000,  
 "growthAbsolute": 3000,  
 "growthPercent": 7.14  
}

### 2. PUT /AddFunds

Description: Adds funds to a user's account.  
Request Body:  
{  
 "user\_id": "string",  
 "amount": "number"  
}  
Response: JSON object indicating success or failure.

### 3. PUT /WithdrawFunds

Description: Withdraws funds from a user's account.  
Request Body:  
{  
 "user\_id": "string",  
 "amount": "number"  
}  
Response: JSON object indicating success or failure.

### 4. GET /GetAll

Description: Retrieves all stocks for a user on a specific date.  
Query Parameters:  
- userId: string  
Response:  
{  
 "stocks": [/\* array of stocks \*/]  
}

### 5. GET /GetPerformanceGraphData

Description: Retrieves historical wealth data for a user over a specified timeframe.  
Query Parameters:  
- userId: string  
- timeframe: number (0, 1, 2, 3)  
Response: JSON array of historical valuations.

### 6. GET /StockPie

Description: Retrieves the distribution of the top stocks in a user's portfolio.  
Query Parameters:  
- userId: string  
Response: JSON array of top stocks.

### 7. GET /SectorPie

Description: Retrieves the distribution of top sectors in a user's portfolio.  
Query Parameters:  
- userId: string  
Response: JSON array of top sectors.

### 8. GET /Balance

Description: Retrieves the user's current balance.  
Query Parameters:  
- userId: string  
Response: JSON object containing balance data.

# Browse API Documentation

Overview:  
The Browse API enables users to access and manage their watchlist, retrieve historical stock data, and get top gainers and losers in the stock market.

## Endpoints:

### 1. GET /Historical

Description: Retrieves historical stock price data for a given ticker and timeframe.  
Query Parameters:  
- ticker: string  
- timeframe: number (0: 1W, 1: 1mo, 2: 6mo, 3: 1y)  
Response: JSON array of historical stock prices.

### 2. GET /watchlist

Description: Retrieves the list of stocks in the user's watchlist.  
Response: JSON array of watchlist stocks.

### 3. DELETE /watchlist/delete

Description: Deletes a stock from the user's watchlist.  
Query Parameters:  
- watchID: string  
Response: JSON object indicating success or failure.

### 4. POST /watchlist/add

Description: Adds a stock to the user's watchlist.  
Query Parameters:  
- stockID: string  
Response: JSON object indicating success or failure.

### 5. GET /top-gainers

Description: Retrieves the list of top gainers in the stock market.  
Response: JSON array of top gainers.

### 6. GET /top-losers

Description: Retrieves the list of top losers in the stock market.  
Response: JSON array of top losers.

# StockInfo API Documentation

Overview:  
The StockInfo API provides detailed company information and allows users to search for stock symbols.

## Endpoints:

### 1. GET /Companyinfo

Description: Retrieves company information for a given ticker.  
Query Parameters:  
- ticker: string  
Response: JSON object containing company information.

### 2. GET /SymbolSearch

Description: Searches for the best matching stock symbols based on a query.  
Query Parameters:  
- ticker: string  
Response: JSON array of matching symbols.

### 3. GET /api/autocomplete

Description: Provides autocomplete suggestions for stock symbols based on a query.  
Query Parameters:  
- q: string  
Response: JSON array of suggestions.

# Checkout API Documentation

Overview:  
The Checkout API handles the purchasing and selling of stocks for a user and retrieves the number of shares owned by a user.

## Endpoints:

### 1. POST /BuyStocks

Description: Records a stock purchase for a user.  
Request Body:  
{  
 "ticker": "string",  
 "share\_quantity": "number",  
 "user\_id": "string"  
}  
Response: JSON object indicating success and transaction details.

### 2. POST /SellStocks

Description: Records a stock sale for a user.  
Request Body:  
{  
 "ticker": "string",  
 "share\_quantity": "number",  
 "user\_id": "string"  
}  
Response: JSON object indicating success and transaction details.

### 3. GET /GetShareAmount

Description: Retrieves the number of shares a user owns for a specific ticker.  
Query Parameters:  
- userId: string  
- ticker: string  
Response: JSON object containing the share amount.