Retirement Planning Tool

Vidya N Kapadia
Bruce T Long
Gargi Sunil Sontakke
Shilpa Chetty Vobbilisetty
Shravani Addagaduru
Satyasaarika Putla

Department of Computer Science, George Mason University
CS 787 - 001: Decision Guidance Systems Final Project
Dr. Alexander Brodsky
December 7, 2022

Section 1: Introduction

Retirement planning is crucial in providing for a person's future financial stability. Among the preparation tools we researched, none provide personalized recommendations to users. To explore solutions to this issue, for our final project, we developed a retirement planning tool, which computes detailed recommendations via three system features. The first feature diagnoses a person's current retirement preparation status; the second recommends a pareto-optimal set of asset allocation weights and contribution amounts when each of the user's accounts are distributed among three types of funds, US Stock, International Stock, and Bonds; the third computes the investment amount required for a user to purchase an annuity at retirement. This report will elaborate on these features in the following order: Section 2 presents an overview of retirement planning in the United States. Section 3 discusses the research done to inspire our system's features. Section 4 presents our system Architecture and utilized technologies. Section 5, 6, and 7 detail each system feature's logic. Section 8 presents limits of our system, challenges faced in development, and potential steps to improve our project. Section 9 reviews lessons we learned through this project. Finally, section 10 presents each Group Member's individual contribution to the project.

Section 2: Overview of Retirement Planning

There are both public and private retirement programs in the United States, and a complex web of account choices and legal requirements. The **public program** includes the Social Security system and is financed by a pay-as-you-go principle. All employed people pay into the system to cover the benefits of the recipients and at the same time build up their own pension entitlements for later. The pension entitlement depends on the payments made and the number of years of contributions. All salaried employers as well as self-employed persons and freelancers are obliged to pay the contributions. In addition to the retirement benefits, Social Security also guarantees access to medical treatment through Medicare, which is a public health insurance program for elderly and disabled citizens[150].

However, the American Social Security system is more of a foundational safety net since it cannot fully maintain the quality of life in old age. As a result, people usually need to supplement their old age pension themselves. For this reason, many tax-advantaged options for private and employer pension plans are offered. Listed below are a few of them:

1) **IRAs:** IRA[4] stands for **Individual Retirement Account**. This is a long-term personal savings account that individuals with an income can use to save for the future and enjoy certain tax advantages. Although intended primarily for self-employed people who do not have access to workplace retirement accounts, the IRA is frequently included in retirement portfolios along with workplace accounts. When an individual opens an IRA they may choose to invest in a wide range of financial products, including stocks, bonds, exchange-traded funds(ETFs), and mutual funds. There are primarily four types of IRAs, depending on different rules regarding eligibility, taxation, and withdrawals:

    a) **Traditional IRA:** allows the user to contribute pre-tax dollars to the retirement account that can grow tax-deferred. The contributions to a traditional IRA may be tax deductible depending on factors such as taxpayer's income, tax-filing status, etc. Unlike Roth IRA contributions, traditional IRA contributions are deductible from the current taxable income. So, if someone's contribution to their IRA is $6000, they can claim that amount as a deduction on their income tax return and the Internal Revenue Service will not apply income tax to those earnings. However, when that individual withdraws money from the account during retirement, earnings are taxed at their ordinary income tax rate.

b) **Roth IRA:** a type of tax-advantaged individual retirement account funded with after-tax dollars. With a Roth IRA, **contributions and earnings grow tax-free** after the age of 59 and a half and can also be withdrawn without any taxes. The biggest distinction between Roth and Traditional IRAs is the way they both are taxed. Roth IRAs are funded with after-tax dollars, meaning the contributions are not tax-deductible, but the investment may grow tax-free. The allowable investments in a Roth IRA includes mutual funds, stocks, bonds, exchange-traded funds(ETFs), certificates of deposit(CDs), money market funds, and even cryptocurrency. This type of IRA is best when the marginal taxes are higher in retirement than they are before it.

c) **Simplified Employee Pension(SEP) IRA:** a type suitable for small business owners and self-employed individuals, where the employer is allowed a tax-deduction for contributions made to a SEP IRA. The contributions are made to each eligible employee's plan on a discretionary basis. The SEP IRAs have a higher annual contribution limit than the standard IRAs. They also can receive employer contributions like 401(k) and are treated like traditional IRAs for tax purposes and allow the same investment options.

d) **Savings Incentive Match Plan for Employees(SIMPLE) IRA:** A SIMPLE IRA is a retirement savings plan that can be used by businesses with 100 or fewer employees. Employers have the choice of making a non-elective contribution of 2% of the employer's salary or a dollar-for-dollar matching contribution of the employee's contributions to plan up to 3% of their salary. This type of IRA is easy to set up, with just an initial plan document and annual disclosures to employees. Employers also get a tax deduction for the contributions they make for employees. One drawback of this type is that the business owner cannot save as much for retirement as with other small business retirement plans, such as the SEP or 401(k).

e) **Rollover IRA:** In this type of IRA, you contribute the money 'rolled over' from a qualified retirement plan into this traditional IRA. Rollovers involve moving eligible assets from an employer-sponsored plan such as a 401(k) or 403(b) into an IRA.[151]

Investing in an IRA can help one to supplement their current savings in the employer sponsored retirement plan. It can also help gain access to a potentially wider range of investment choices than the employer-sponsored plan. A person investing in an IRA can also take advantage of potential tax-deferred or tax-free growth.

2) **401(k):** a retirement **savings plan** offered by many American employers that allows employees to save and invest a portion of their paychecks before tax deductions. This is a company-sponsored retirement account to which both the employers and the employee contribute. With a 401k plan, a certain percentage of each paycheck is paid into this investment account and the employer may match part or all of that contribution. The 401(k) also has two categories based on how they are taxed:

a) **Traditional 401(k):**In the traditional 401(k), the employee contributions are deducted from the gross income meaning the money deducted comes from the employee's payroll before the income taxes have been deducted. No taxes are due neither the money contributed or the investment earnings until the employee withdraws the money, usually in retirement. For example, if you earned $80,000 in 2021 and contributed $5000 towards towards 401(k), your taxable income will be reduced to $75,000. However, when you withdraw from your account in retirement, your contributions and investment earnings are generally fully taxable.

b) **Roth 401(k):** In the Roth 401(k), contributions towards the retirement account are deducted from the employee's after-tax income. As a result, there is no tax deduction in the year of the year of contribution. When the money is withdrawn during retirement, no additional taxes are due on the employee's contribution or the investment earnings.

The primary difference between a 401(k) and an IRA is that an employer may offer a participant a 401(k), whereas individuals may purchase their own IRA. IRAs do not offer benefits like employer matching or a higher contribution limit, but they provide participants more flexibility and investment choices than a 401(k) can.

3) **HSA**: HSA stands for **Health Savings Account,** which puts aside pre-tax dollars to be used on medical expenses, prescriptions or medical equipment. By using the untaxed dollars in a Health Savings Account to pay for deductibles, coinsurance and some other payments, one can lower their overall health care costs. However, an HSA is available only to individuals covered under high-deductible health plans(HDHP). A benefit of the HSA is that, while owned by an employee, it can be funded by the employer. HSA contributions may be invested, and unused account balances at year-end can be carried forward.[152]-[153]

4) **Brokerage**: an individual account used to buy and sell securities such as stocks, bonds, mutual funds and ETFs. It gives the user direct access to the stock market and other related investments. Brokerage accounts are also called taxable investment accounts to differentiate them from tax-advantaged retirement accounts like 401(k)s. They can be used for day trading to earn short term profits as well as investing for long term goals.[154]

5) **ETFs and mutual funds**: ETFs are **Exchange Traded funds**, a pool or basket of securities that trades on an exchange just like a stock does. It can contain all types of investments including stocks, commodities or bonds and can either be actively or passively managed. ETFs trade like stocks and are primarily passive investments that seek to replicate the performance of a particular index. Passive ETFs tend to be tax efficient, in part because tracking an index usually does not require frequent trading.[155]-[156], [35]

   Mutual funds are generally bought directly from investment companies instead of from other investors on an exchange. Unlike ETFs, they do not have trading commissions, but they do carry an expense ratio due to management and sales fees. Like ETFs, mutual funds may be actively or passively managed. Index mutual funds are considered passive investments because they mirror an index, but their performance may beat some but not all actively managed funds. The investments in an actively managed mutual fund are selected and managed by a portfolio manager.

   Mutual funds and ETFs have a lot in common. Both types of funds consist of a mix of many different assets and represent a popular way for investors to diversify. However, there are also a few key differences between the two, including the way they are traded. ETFs may be traded intraday like stocks, while mutual funds only can be purchased at the end of each trading day based on a calculated price known as the net asset value.

   Our model is based on Vanguard's **standard** three fund portfolio, separating investments into simplified asset categories of domestic and international stocks and bonds, for these reasons: First, it **simplifies diversification,** which helps in managing risks better. It also reflects the market as a whole, emphasizing the major categories of funds. Managing one’s own allocation of investments to the different fund categories will cost less than actively managed funds. Finally, because they are easy to invest in, even a beginner in investments has less to worry about.

Section 3: Inspiration for our System/Existing State of the Art

The inspiration for our system is the three-fund portfolio which is a simple retirement strategy that is employed by most major brokerages’ Target Date Funds ([88]) and championed by the Bogleheads. (According to [36, 37], the Bogleheads are a group of fans of John Bogle, the founder of Vanguard and creator of the first index fund). Specifically, according to [38], the portfolio consists of three diverse, passively managed funds (as they incur low cost):, one domestic stock, one international stock and one bond fund. We allow our user to select one fund for each of these categories.

We drew some inspiration for our features (including the structure of our GUI, and backend functionality) from a few current retirement calculators that are out there. We looked at Personal Finance Club's Investment Growth Retirement Calculator ([70]) which provides users with an estimate of what they can expect their portfolio to grow to in retirement taking into account their cost of living, but it doesn't provide the user with any actionable information.

Another calculator that provided inspiration for our project is cFIREsim ([71]), a simulator based on historical data that provides the user with hypothetical scenarios of how their portfolio would have performed with several historical start dates. This calculator provided more information around the potential risks associated with investing in the stock market, but is overly complex for most users.

The ADP Growing Your Investment calculator ([89]) is another simple calculator that visually demonstrates the benefits of compound interest by separating the growth/interest from the contributions. Other calculators we reviewed include Fidelity's Contribution Calculator ([90]) and Bankrate's 401k ([91]) and Asset Allocation ([92]) Calculators.

We also were inspired by Vanguard's Investor Questionnaire ([97]), which asks users a series of hypothetical questions to determine their personal risk tolerance and provides an optimal asset allocation for that user. However this questionnaire does not take into account a user's individual spending and time horizon.

Our goal was to have a low barrier of entry by asking users for simple information that they would easily be able to provide, then create recommendations that are more actionable and specific to the individual user.

Section 4a: Process Followed to Construct Our System

To build this decision guidance system, we initially performed several steps:

- We researched the retirement domain and various calculators (refer to citations document for a detailed description of sources we consulted in our research)
- We brainstormed for ideas and system features
- We formed diagnosis code
- We formed an initial model of system
- We sought feedback from our professor

We then followed the process outlined below through multiple iterations over the course of several weeks to improve our model and system code:

1. Conduct further research as needed
2. Expand and correct model
3. Expand and correct computer code
    a. Backend processing
    b. User interface
4. Perform data input and testing
5. Solicit feedback from users and professor

Although we increased the amount of time spent on improving our user interface as the system neared completion, we were still tweaking the model and improving the backend up to shortly before the reporting date.

Section 4b: System Architecture and Tools Used

Our application includes a Graphical User Interface (GUI), server, and backend. We implemented the GUI with the following tools:

Front end:
- HTML
- CSS/BOOTSTRAP [98] - [101]
- JAVASCRIPT
- CHART.JS [102] - [106]
- GOOGLE CHARTS [107] - [111]

Backend:
- PYTHON
- FLASK FRAMEWORK FOR REST API[82]

Model:
- DGAL
- PYOMO

Cloud:
- HEROKU- For backend deployment[81]
- AWS EC2- For frontend hosting[79],[148]

We implemented our server with Flask, and our backend logic with other Python libraries, including Decision Guidance Analytics Language (DGAL) ([159]), a module created by Professor Brodsky which we practiced utilizing in homework assignments. Additionally, we deployed our server and backend on Heroku and our GUI on AWS EC2. **Please see Appendix A for a detailed explanation of the sources we referenced to construct our code with these tools.**

These system components interact in the following order: our GUI presents a form for user's to input their retirement account information. On submission, it forwards user input via fetch() Javascript functions to API routes on our server. These routes invoke our backend code to compute and return results for our three features. Finally, our GUI presents these results to the user. The following sections will explain the implementation logic of each feature in our backend.

Section 4c: System Inputs

System Inputs:
- Current Income
- Current Spending
- Current Taxes
- Account Information (see below)
- Retirement Age
- Life Expectancy
- Expected Spending in Retirement
- Expected Taxes in Retirement

- Expected Social Security Benefit Amount
- Expected Pension Payments
- Inflation and Salary Increase Expectations

For each account:

- Account Type
- Current Contributions to Account
- Current Account Balance
- Current Asset Allocation
- Current Account Rate of Return

Section 4d: Supporting Data Sources

etf_options.json

- Includes all ETF options for our system
- Includes the type of each fund, US/Intl/Bond
- Provides Expense Ratio and Inception Date
- Has backup expected returns information

IRSLimits.json

- Stores the current IRS limits, both regular and catch up for 401ks, HSA, and IRAs
- Also includes a breakdown of how each account is taxed, for future tax implementations.
- Used to check contribution limit constraints

table_iii.json

- Stores the current IRS Uniform Lifetime Table III
- Used in the calculation of RMDs for Traditional IRAs

Section 4e: System Outputs

System Outputs:

- Diagnosis of whether the user's savings will last through retirement
- Recommendation 1: Set of points representing different pareto-optimal asset allocations and optimal contribution amount according to user's budget
- Recommendation 2: The investment amount user would need to deposit to purchase an annuity at retirement and whether their account balance covers this amount

Section 5: Diagnosis Feature

In order to determine the user's current financial situation and whether they are on track financially to make it through retirement successfully, we take their inputs and run it through a diagnosis model. The user provides their current income, spending, taxes, contributions and account information as well as an estimate of their year of retirement, retirement spending, taxes in retirement and additional income in retirement. We also ask the user to input their expectations around account growth, inflation and salary increases. Our diagnosis code takes this information and checks that their accounts, contributions and growth is enough to sustain their spending and taxes through retirement. If it is not, we return to the user the age until their current spending will last them.

To ensure we are taking into account Required Minimum Distributions (RMDs) when the user turns 72, we calculate the RMDs for each year beginning at age 72, deduct that amount from their Traditional IRA and then add that amount to their income for that year. We calculate the RMDs based on the Uniform Lifetime Table (Table III [149]) from the IRS.

Section 6: Optimal Asset Allocation and Contributions Feature

After diagnosing the user's financial status, if their current salary is greater than their spending and tax amounts, our backend computes a range of pareto-optimal asset allocations and contributions in the following manner. First, our backend uses user input to construct a mathematical model representing the user's retirement account portfolio. Afterwards, it invokes the DGAL engine which retrieves results (to construct a set of pareto optimal points) by running the linear optimizer, GLPK, on a Pyomo representation of our model. Section 6a details the construction of our model. Section 6b discusses the algorithm used to produce a pareto optimal front.

Section 6a: Model

According to course instruction and the Operations Research perspective ([55, Ch. 1, p. 1], [55, Ch.3, pp. 49-53], [55, Ch. 1, p. 5]) a mathematical model must represent an actual system in the form of mathematical equations which specify metrics and constraints on decision variables. Applying this knowledge, the professor's feedback, and the model format from class assignments, we created two models: single account model and combined account model (these models' implementations are in functions, singleAccountMod and combiningAccountsMod, respectively, in our modelcode.py file). Each single account model represents a different retirement account. When combined, these models' outputs form metrics and constraints for one combined account model, which represents the user's retirement account portfolio.

We formed each model's metrics and constraints to represent the domain constructs on actual retirement accounts. In general, the user's goal is to maximize their portfolio's balance by retirement (to ensure they are taking maximum advantage of the retirement accounts available to them and to maintain their savings until death). And as noted in section 5, every retirement account's balance increases when the rate of return and contribution amount (deposited by our user) are large. As shown in the portfolio expected return formula in Table 2, to produce higher return rate for their account, the user can allocate more weight to securities with higher expected return, such as exchange trade and mutual funds that invest in the total stock market ([32]-[35] and [40] give more information on these and other types of ETFs and mutual funds). However, higher return securities undergo higher volatility, which [61] defines as the variation in returns. Thus, a higher return may decrease drastically, producing losses for the user's account over time. Additional losses on the account are incurred if the user invests in funds with high expense ratios (which charge high maintenance fees) (according to [32]-[35], [40], [42], [43], [45]). As for contributions, the Internal Revenue Service (IRS) sets yearly legal limits for these amounts in different types of retirement accounts, based on the user's current age category (below 50 and greater than 50 years old) ([25]-[29], [87]). Thus, the user can only contribute up to this yearly amount. Furthermore, to maintain financial stability, these contributions must not exceed the user's annual income.

Applying this domain knowledge, we derived the following variables, metrics, and constraints to represent our model. First, we identified decision and fixed variables for our models. The user's decision variables (which directly influence their objective to maximize account balance by retirement) include weights of securities each account is distributed amongst, and the yearly contribution amount to each account. The fixed variables in our model (which encode the user's current situation) include the user's pre retirement salary, spending, and taxes, years until the user retires (called X), the expected return and standard deviation over the X year period for each chosen security (US Stock, International Stock, and Total Bond funds) in every inputted retirement account, and legal limits on contribution amounts.

Afterwards, we formed metrics and constraints in terms of these variables for each model. The single account model computes the main metrics: portfolio expected return, portfolio standard deviation (which is a common metric for volatility according to [61],[150], among other risk metrics such as beta and sharpe ratio ([63])), total account balance by retirement, and total cost incurred on the account by retirement. The single account model constraints limit the weights of each security in the account to be between 0 and 1, and the yearly contributions to be within a user's budget and IRS legal limits. As for the combined account model, the model computes totals of the metrics outputted from single account model and adds a constraint limiting the total yearly contribution to all accounts to be within the user's budget. **Tables 1, 2, and 3 specify our models' inputs and outputs (metrics and constraints). Please see table 2's description column for sources we referenced to calculate each metric in our code.**

Besides these primary metrics, our models compute linear proxies representing the growth of each retirement account and the standard deviation. We use these proxies in objective functions to produce our pareto optimal front of asset allocations, which we detail in the next section.

Table 1: Inputs to Single and Combined Account Models (below):

Model	Inputs with Descriptions
Single Account Model	● Account Information ○ Account Type (valid domain: Traditional (Trad) and Roth IRAs, 401ks, HSAs, Brokerages, or all (for one account, if the user does not wish to enter information for all accounts)) ○ Account Asset Information: consists of the three funds the user chooses to distribute the account among, and their weights ○ Total yearly investment contribution ○ Current balance in the account (as of day of input) ● User Information: ○ Current annual income ○ Current annual spending ○ Current annual taxes ○ Retirement age ○ Range to run model on (to compute growth in accounts over the range, this is the range from current to retirement year)
Combined Account Model	● Account Type (value: combined) ● Sub Accounts: a list of all retirement account types inputted by the user ● Sub Account Inputs: corresponding single account model input for each account type listed in Sub Accounts (this model invokes the single account model code with each input and uses the results to compute its metrics and constraints on the user's retirement account portfolio)

Table 2: Metrics for Single and Combined Account Model (below):

Model	**Metric**	**Metric Formula/Description**
Single Account Model	Portfolio (Account) Expected Return	Portfolio Expected Return Formula (used from [56]):

		$\sum_{i=0}^{i=3} w_i E_i$, where w_i = weight of security (inputted to the model/found in optimization) E_i = security's annualized return rate Each *Ei* is computed according to the annualized return rate formula (from [131] and [132]) as shown below: $Y^{1/z} - 1$, where z = X years to retirement Y = average return over X (years to retirement) period from days within range: inception date of the fund, j, to the current day, k, in the formula below: $\mathrm{Y} = \frac{\sum_{i=j}^{i=k} \frac{adjusted\ close\ price\ in\ day(\ i + X * 252)}{adjusted\ close\ price\ in\ day\ i}}{k - j}$ We calculated Y according to the Professor's feedback. Additionally, the days to retirement = X * 252, since there are 250 to 252 days a year in which fund's shares are traded according to [57,58]. Also, we use Pandas DataReader to scrape the adjusted close prices used in Y's calculation from Yahoo Finance's website (as done in sources [57,58, 133]). If days to retirement is greater than the inception date of a fund (ie, X * 252 > (k - j)), we calculate Y using the upper bound i = (k - j)/2 (to avoid errors with calculating standard deviation for the security, as explained below) (**Note: This expected return calculation varies from what was presented in our slides.** Specifically, we initially attempted to use methods/syntax from [57],[58],[59],[60],[133] to calculate annualized return rate for each Ei with geometric mean formula (from [59],[60]) using the annual rates (from

		close prices) in range: [inception date of fund i, current date]. However, the Professor helped explain that this method did not account for the time in which the user will keep their investment in these funds (years to retirement). Afterwards, he guided us to use the strategy above.)
	Portfolio Standard Deviation	We applied the formula from [62] shown below, to calculate Portfolio Standard Deviation: $\sqrt{\sum_{i=0}^{i=9} w_r w_c s_r s_c cor(r,c)}$ for all combinations of the three funds securities (9 possible) in the account. where : w_r and w_c = weights of fund r and c $cor(r,c)$ = correlation of returns over each X year period(X = years to retirement) for fund r and c s_r and s_c = standard deviations of fund r and c. Each fund i's standard deviation, s_i is computed using the formula below which is derived from ([134],[135]) and the Professor's feedback: $\sqrt{\frac{\sum_{i=j}^{i=k} (a-Y)^2}{X}}$, where a = return over one X year period, Y = average for all returns in X year period (calculated in portfolio expected return formula), X = years to retirement, j = date of inception, k = current date or (k - j)/2 if X > (k -j)
	Total Account Balance By Retirement	We used the compound interest formula to calculate account balance each year until retirement like so: $\sum_{i=t}^{i=retyr} balance + growth - cost$ where t = current year, $retyr$ = retirement year $balance$ = account balance at end of

		year i - 1 $growth$ = ((balance * portfolio expected return) + yearly contribution) $cost$ = annual maintenance fees for all securities (from expense ratios), in detail: ([158]) $cost = \sum_{i=1}^{i=3} r\, wbalance$, where r = expense ratio for security i, w = weight of security i in the account
	Total Cost Incurred on the Account By Retirement	$\sum_{i=t}^{i=retyr} cost$, where $retyr$ = year of retirement, t = current year, $cost$ = value computed above in total account balance by retirement formula
	Total Growth of the Account by Retirement (Proxy)	L * ((T * R) + C), where L = retirement year - current year, T = current investment account balance R = portfolio expected return C = yearly contribution to the account
	Standard Deviation of the Account (Proxy)	$\sum_{i=0}^{i=3} w_i Std_i$,where Std_i = standard deviation of fund i (calculated as s_i in Portfolio standard deviation formula) w_i= weight of the fund in the portfolio i = 3 upper bound since there are three funds that consist the retirement account (US, International Stock, and Bonds)
	Total account balance by retirement minus std dev	This metric calculates the total account balance with portfolio expected return decremented three std deviations

	Total account balance by retirement plus std dev	This metric calculates the total account balance with portfolio expected return incremented three std deviations
Combined Account Model	Totals of each metric computed in single account model (example are the proxies listed below)	Sum of metric values for all retirement accounts inputted into GUI
	Total Growth of All Accounts by Retirement (Proxy)	Sum of each single retirement account's metric: Total Growth of Account by Retirement (Proxy)
	Total Standard Deviation from all Accounts (Proxy)	Sum of each single retirement account's metric: Standard Deviation from Accounts (Proxy)

Table 3: Constraints for Single and Combined Models

Model	Constraints
Single Account Model	• Weight >= 0, for every asset in the retirement account • Sum of weights of each asset = 1 • Yearly contribution >= 0 • Yearly contribution < Legal (IRS) yearly limits for user's age • Age legal limits: after 65, the contribution amount to HSA account is 0, and after retirement the user cannot contribute to their 401k (this clause does not execute since our model is only run til retirement) • Yearly contribution within Budget: yearly contribution to the retirement account < salary - spending - taxes • Allocate more according to Traits: According to the Trinity study ([72]), users who spend less than 4% of their savings each year will not run out of funds by death. Additionally, sources (cite [39],[136]) suggest that users closer to retirement should shift more of their account into bonds. Thus, we initially planned to implement this constraint accordingly, placing higher weights to bond funds if the user is closer to retirement, and stock funds otherwise. However, we ended up not using this constraint, since the pareto optimal front would allow users to choose their asset allocation
Combined Account	• Yearly contributions to all accounts within Budget : (sum of the

Model	contributions in each single account models is less than salary after spending and taxes), ● subAccountConstraints: all constraints for each single account model (representing a retirement account in the user's portfolio), are fulfilled

Section 6b: Pareto Optimal Front

As implied in section 6a, there is a direct linear relation between standard deviation (volatility) and expected return. In general, risk averse users (seeking to preserve their investment) are willing to compromise expected return for a lower standard deviation in their account. Thus, financial advisors ([136],[137]) recommend users allocate more of their accounts into set proportions of stocks and bonds based on their level of risk. Additionally, these sources([39],[136],[137]) recommend that to maximize retirement account balance, users start with a high stock allocation in their youth and allocate more towards bonds as they approach retirement.

In order to balance this financial advice with the user's preference for risk within their current situation, our system allows the user to select their asset allocation from a pareto-optimal range (i.e., where only one metric (standard deviation or expected return) has an optimal value). Specifically, after the user completes form submission, our GUI presents up to 5 (X,Y) coordinates (computed in our backend), where each X coordinate corresponds to the cumulative expected return on the user's retirement account portfolio, and each Y coordinate corresponds to the account balance accumulated by retirement for the expected return represented by X. Additionally, our graph displays minimum and maximum values for the Y coordinate, which correspond to the account balance accumulated by retirement when expected return varies by three standard deviations. Thus, our graph displays a 99% confidence interval of account balance by retirement for each point.

To compute these points, our backend utilized the paretoOptimal algorithm given by Professor Brodsky and derived from his paper ([138]). The following explains the application of this algorithm to compute our points. First, our system utilizes DGAL to derive the minimum and maximum values for the two metric proxies described in Table 2 for the combined account model: **Total Growth of all Accounts by Retirement (Proxy)** and **Standard Deviation of all Accounts (Proxy)**. Our code sets these linear proxies as objective functions to incorporate all decision variables in each retirement account model and utilize GLPK. (Thus, our system runs DGAL 4 times, with each objective function, objective (if minimizing the objective function, dgal.min() executes, else dgal.max() executes), and the combined model). Afterwards, our system computes each point by running DGAL (the max() function) with the objective function below (which combines our normalized proxies) with a different angle. So, to compute 5 points, our system runs DGAL 5 more times: for point 1, *angle*= (1* (89))/5, for point 2: *angle* = (2* (89))/5, and for point 5: *angle* = (5 * (89))/5) :

$X_{normalized}$ cos(*angle*) + $Y_{normalized}$ sin(*angle)*,
Where:

$X_{normalized}$= (Total Growth of all Accounts by Retirement Proxy - min X) /(max X - min X)

$Y_{normalized}$ = (total standard deviation of all accounts proxy - min Y)/(max Y - min Y) ,

min Y, max Y, min X, and max X are the optimal values for the proxies
(the minimum and maximum values computed when the objective function is one proxy,
which is computed in the first part of the algorithm)

angle = (i * (90 - 1)/n), where i = point number, n = total number of points generating on the graph **(the angle is computed like so to find coordinates of all n points distributed within the 90 degree angle space)**

Each optimization takes 20 seconds to complete. So, to avoid timeout errors we distributed these calls for optimization among different api routes in our server, which our GUI uses to fetch this data..

Section 7: Annuity Feature

Another option we provide to our users is annuities, which is an ideal alternative to choosing asset allocations for risk averse users, due to their low returns and risk ([12], [50]-[53], [93]-[97]). As for computation, based on our user's expected spending in retirement, we calculate the amount needed to purchase an annuity that would provide that level of consistent spending throughout retirement (similar to the function of the Brighthouse Financial Calculator ([54])). We used the Payout Annuity Formula from annuity.org ([69]) to calculate this principal. The picture below is from [69], r = return rate (we set it at the average which is 2% according to ([139]), n = 12 (for monthly payments), t = years until death, PMT is the monthly payout of the annuity, which we set to the user input of expected annual spending in retirement divided by 12 to get the monthly expected spending in retirement. This ensures that the annuity purchased will cover the user's spending in retirement.

$$P_0 = \frac{PMT\left(1-\left(1+\frac{r}{n}\right)^{-nt}\right)}{\frac{r}{n}}$$

Section 8: Limits, Challenges, and Future Steps

- Many of our funds do not have underlying indices, so to maintain consistency we decided to use the prices of funds instead. However, this still produces inconsistent results when inception dates are later than years to retirement (when there's not enough data for calculating expected return for years of retirement windows). An option for future changes would be to calculate the expected return based on the underlying index for funds that follow a specific index.
- The IRS changes contribution and RMD limits each year. We did not want to make an estimate of future increases to these limits so our model assumes the latest published year (currently 2023) limits continue into the future.
- The Pandas DataReader for Yahoo Finance that we used to get the current and historical prices of the ETFs does not have inception dates or expense ratios, so we created a separate database with this info.
- The Pandas DataReader for Yahoo Finance was also missing some daily close prices since inception, which caused us to incorrectly calculate expected return.
- Inflation is difficult to predict, we are allowing the user to input their expectation of inflation as part of the calculation.
- Expense ratios of ETFs and Mutual Funds in 401ks are negotiated by each employer, so there is no consistent source for this information. We are using the expense ratio of the matching retail available ETF.
- We also wanted to use a non linear solver. We tried using IPOPT, it would output an "optimal" solution, but some of the constraints returned as false. We could not find another non linear solver that would work and our troubleshooting with IPOPT did not return a satisfactory solution, so we kept our model linear with proxy calculations.
- One obstacle we struggled with was how to obtain a user's risk tolerance level. We initially discussed using a version of the Vanguard questionnaire ([97]), but ultimately (after Professor Brodsky's feedback) decided on demonstrating risk using the user's own data using a pareto optimal front to show the user the impact of

various asset allocations on their portfolio. This allows the user to evaluate their own level of risk using a more applicable example.

- Future extensions of this project could include adding accurate tax treatment of the accounts, building a more granular breakdown of spending for the user to allow them to be more precise in their estimates, adjusting asset allocation over time, adapting the UI to more precisely show the impact of compound interest and fees on their portfolio, and providing other types of assets to include as part of the portfolio. Other possible recommendations we could make to a user could include reverse mortgages or reducing spending.

Section 9: Lessons Learned

In order to build an accurate Decision Guidance System, we learned that you need a lot of domain knowledge and access to accurate data. We encountered obstacles in obtaining accurate and accessible data sources so we had to recalculate some statistics from raw data. Working on this project also taught those on our team with little domain knowledge in this area the importance of saving for retirement early to take advantage of compound interest.

Section 10: Group Member Contributions

Each Group Member created parts of the presentation, report, and citation of sources based off of their individual research. Additionally, each group member assisted each other with debugging code. Individual Contributions are listed below:

Vidya - Designed model, provided domain knowledge, implemented parts of diagnosis/model code, provided real life sample input examples, contributed to presentation and final report document. Vidya also defined implementations for domain specific metrics and constraints in the code and helped Saarika debug the model and diagnosis code.

Saarika - Assisted Vidya with creating the model and translated it into code. Specifically, implemented and tested the following code for our system: our diagnosis code (personWillBeOkCode.py, using pseudocode defined by Vidya), modelcode.py file, the server code (api routes) in main.py (and fetch calls in our script.js), and functions to run the optimizer and package system/model output in packingFunctions.py. Also, configured and deployed this backend code to Heroku. Additionally, led team meetings each week and consulted with Vidya to allocate tasks to group members. Finally, organized citations document for the project.

Shravani -
1. Worked on researching the architecture and technologies that best suit our project. I gave an overview of all the technologies and architectures we could follow during our team meetings and together we finalized to go with REST.
2. Worked on the front end part (Html, CSS, bootstrap, javascript) to create an input form that takes users' input and posts the data to the back end.
3. Major portions of my part work worth pointing out would be account type implementation, its validations on the client side, API fetch calls, functionality, and usability testing.
4. Participated in team meetings to discuss model-based implementations of account types, inputs, objectives, and features to show as output to the user.
5. Also, worked on deploying/hosting our website to AWS EC2.

Shilpa - Designed and developed a front-end interface to show recommendations to users using HTML, CSS, Java Script, and Bootstrap and visualized the model's output via Pi, Line, and Candlestick charts.

Gargi - Researched the different resources for domain knowledge such account types for retirement. Researched the different possible options for a potential front-end framework, tried and tested initial test runs on those choices and discussed with the team to finalize the stack. Helped design and develop front end like the input form, including user interface.

Bruce - Identified resources for domain knowledge, investment options, legal requirements and other constraints. Supplied fund performance and risk data, required minimum distributions, and contribution limits. Provided oversight of team's development of model and code. Assisted with creation of presentations for class and professor. Helped write and edit final report.

References

[1] "Sources of Retirement Income | FINRA.org." https://www.finra.org/investors/learn-to-invest/types-investments/retirement/managing-retirement-income/sources-retirement-income (accessed Sep. 01, 2022).

[2] A. Fontinelle, "Reverse Mortgage Definition," *Investopedia*. https://www.investopedia.com/mortgage/reverse-mortgage/ (accessed Sep. 05, 2022).

[3] A. Twin, "Refinance: How and When It Happens," *Investopedia*. https://www.investopedia.com/terms/r/refinance.asp (accessed Sep. 05, 2022).

[4] The Investopedia Team, "What Is an IRA?," *Investopedia*. https://www.investopedia.com/terms/i/ira.asp (accessed Sep. 05, 2022).

[5] Liz Manning, "Federal Insurance Contributions Act (FICA)," *Investopedia*. https://www.investopedia.com/terms/f/fica.asp (accessed Sep. 01, 2022).

[6] Julia Kagan, "How Social Security Benefits Work," *Investopedia*. https://www.investopedia.com/terms/s/social-security-benefits.asp (accessed Sep. 05, 2022).

[7] J. Lee, "What Is Home Equity And How Does It Work?," *Bankrate*. https://www.bankrate.com/home-equity/what-is-home-equity/ (accessed Sep. 05, 2022).

[8] J. Brown, "What Is A Home Equity Loan?," *Bankrate*. https://www.bankrate.com/home-equity/what-is-home-equity-loan/ (accessed Sep. 05, 2022).

[9] The Investopedia Team, "What Is a Pension?," *Investopedia*. https://www.investopedia.com/terms/p/pensionplan.asp (accessed Sep. 05, 2022).

[10] The Investopedia Team, "Defined-Benefit vs. Defined-Contribution Plan: What's the Difference?," *Investopedia*. https://www.investopedia.com/ask/answers/032415/how-does-defined-benefit-pension-plan-differ-defined-contribution-plan.asp (accessed Sep. 05, 2022).

[11] Adam Hayes, "What Is a Lump-Sum Payment?," *Investopedia*, Dec. 29, 2021. https://www.investopedia.com/terms/l/lump-sum-payment.asp (accessed Sep. 05, 2022).

[12] Julia Kagan, "What Is an Annuity?," *Investopedia*. https://www.investopedia.com/terms/a/annuity.asp (accessed Sep. 08, 2022).

[13] Claire Boyte-White, "How Does a 401(k) Work After Retirement?," *Investopedia*. https://www.investopedia.com/articles/personal-finance/111615/how-401k-works-after-retirement.asp (accessed Sep. 16, 2022).

[14] Kelley C. Long, "Medicare's tricky rules on HSAs after age 65," *Journal of Accountancy*, Jul. 01, 2021. https://www.journalofaccountancy.com/issues/2021/jul/medicare-rules-on-hsa-after-age-65.html (accessed Sep. 16, 2022).

[15] Amy Fontinelle, "Retirement Uses for Your Health Savings Account (HSA)," *Investopedia*. https://www.investopedia.com/articles/personal-finance/091615/how-use-your-hsa-retirement.asp (accessed Sep. 16, 2022).

[16]"5 ways HSAs can help with your retirement | Fidelity," Aug. 22, 2022. https://www.fidelity.com/viewpoints/wealth-management/hsas-and-your-retirement (accessed Sep. 05, 2022).

[17] "How do deductibles, coinsurance and copays work? | bcbsm.com." https://www.bcbsm.com/index/health-insurance-help/faqs/topics/how-health-insurance-works/deductibles-coinsurance-copays.html (accessed Sep. 16, 2022).

[18] Davalon, "How a Deductible Works for Health Insurance." https://www.ehealthinsurance.com/resources/affordable-care-act/deductible-insurance (accessed Sep. 16, 2022).

[19] "Health Insurance Costs Like Premiums & Deductibles | Medical Mutual." https://www.medmutual.com/for-individuals-and-families/health-insurance-education/health-insurance-basics/understanding-costs.aspx (accessed Sep. 16, 2022).

[20] Les Masterson, "What Is A Health Insurance Deductible? – Forbes Advisor." https://www.forbes.com/advisor/health-insurance/what-is-a-health-insurance-deductible/ (accessed Sep. 16, 2022).

[21] A. E. M. Hess, "Seven ways Americans pay taxes," *USA TODAY*, Jan. 04, 2014. https://www.usatoday.com/story/money/personalfinance/2014/01/04/taxes-americans-pay/4307825/ (accessed Sep. 17, 2022).

[22] "IRA Required Minimum Distributions Table 2022," *SmartAsset*, Aug. 23, 2022. https://smartasset.com/retirement/rmd-table (accessed Sep. 17, 2022).

[23] "Taxation of required minimum distributions | Vanguard." https://investor.vanguard.com/investor-resources-education/taxes/taxation-of-required-minimum-distributions (accessed Sep. 17, 2022).

[24] "Benefits Planner | Income Taxes And Your Social Security Benefit | SSA." https://www.ssa.gov/benefits/retirement/planner/taxes.html (accessed Sep. 17, 2022).

[25] "Retirement Topics - IRA Contribution Limits | Internal Revenue Service." https://www.irs.gov/retirement-plans/plan-participant-employee/retirement-topics-ira-contribution-limits (accessed Sep. 17, 2022).

[26]"401(k) Contribution Limits for 2022 – Forbes Advisor." https://www.forbes.com/advisor/retirement/401k-contribution-limits/ (accessed Sep. 17, 2022).

[27]"Retirement Topics - 401(k) and Profit-Sharing Plan Contribution Limits | Internal Revenue Service." [Online]. Available: https://www.irs.gov/retirement-plans/plan-participant-employee/retirement-topics-401k-and-profit-sharing-plan-contribution-limits. [Accessed: Sep. 20, 2022]

[28] S. Miller, "IRS Announces Spike in 2023 Limits for HSAs and High-Deductible Health Plans," *SHRM*, Apr. 29, 2022. [Online]. Available: https://www.shrm.org/resourcesandtools/hr-topics/benefits/pages/2023-irs-contribution-limits-for-hsas-and-high-deductibel-health-plans.aspx. [Accessed: Sep. 20, 2022]

[29] B. Ruane, "Rev. Proc. 2021-25." IRS [Online]. Available: https://www.irs.gov/pub/irs-drop/rp-21-25.pdf. [Accessed: Sep. 20, 2022]

[30] IRS, "IRA Required Minimum Distribution Worksheet." IRS, 2022 [Online]. Available: https://www.irs.gov/pub/irs-tege/uniform_rmd_wksht.pdf

[31]"Retirement Plans FAQs regarding Required Minimum Distributions | Internal Revenue Service." [Online]. Available: https://www.irs.gov/retirement-plans/retirement-plans-faqs-regarding-required-minimum-distributions. [Accessed: Oct. 04, 2022]

[32] "Types of ETFs," *Schwab Brokerage*. https://www.schwab.com/etfs/types (accessed Sep. 27, 2022).

[33] "Vanguard ETF/fund ratios - Bogleheads." https://www.bogleheads.org/wiki/Vanguard_ETF/fund_ratios (accessed Oct. 04, 2022).

[34]"Types of mutual funds," *Schwab Brokerage*. https://www.schwab.com/mutual-funds/types (accessed Sep. 27, 2022).

[35] "ETFs vs Mutual Funds," *Schwab Brokerage*. https://www.schwab.com/etfs/mutual-funds-vs-etfs (accessed Sep. 27, 2022).

[36] "The Bogleheads® - Bogleheads." [Online]. Available: https://www.bogleheads.org/wiki/The_Bogleheads%C2%AE . [Accessed: Oct. 04, 2022]

[37] Bogleheads, "Bogleheads® investment philosophy," Bogleheads® wiki, Mar. 2009. [Online]. Available: https://www.bogleheads.org/wiki/Bogleheads%C2%AE_investment_philosophy. [Accessed: Oct. 04, 2022]

[38] Bogleheads, "Three-fund portfolio," *Bogleheads Wiki*, Nov. 06, 2010. [Online]. Available: https://www.bogleheads.org/wiki/Three-fund_portfolio . [Accessed: Oct. 04, 2022]

[39] J L Collins, "Stock Series," J L Collins The Simple Path to Wealth, 2022. https://jlcollinsnh.com/stock-series/ (accessed Oct. 04, 2022).

[40] "What Is an Exchange Traded Fund (ETF)?," *Investopedia*. https://www.investopedia.com/terms/e/etf.asp (accessed Sep. 22, 2022).

[41]"Why Market Cap Matters - Fidelity." https://www.fidelity.com/learning-center/trading-investing/markets-sectors/why-market-cap-matters (accessed Sep. 22, 2022).

[42] "VOO vs. VTI - Vanguard's S&P 500 and Total Stock Market ETFs," https://www.optimizedportfolio.com/. https://www.optimizedportfolio.com/voo-vs-vti/ (accessed Sep. 22, 2022).

[43] "The 5 Best Index Funds with Low Expense Ratios You Can Invest in Right Now," *Time*, Oct. 27, 2021. Accessed: Sep. 27, 2022. [Online]. Available: https://time.com/nextadvisor/investing/5-best-index-funds-low-expense-ratio/

[44] "Charles Schwab vs. Vanguard," *Investopedia*. https://www.investopedia.com/charles-schwab-vs-vanguard-4587941 (accessed Sep. 27, 2022).

[45] "Best S&P 500 ETFs for Q4 2022," *Investopedia*. https://www.investopedia.com/investing/top-sp-500-etfs/ (accessed Oct. 04, 2022).

[46] "VTI-Vanguard Total Stock Market ETF | Vanguard." https://investor.vanguard.com/investment-products/etfs/profile/vti (accessed Oct. 04, 2022).

[47] "VOO-Vanguard S&P 500 ETF | Vanguard." https://investor.vanguard.com/investment-products/etfs/profile/voo (accessed Oct. 04, 2022).

[48] "BND-Vanguard Total Bond Market ETF | Vanguard." https://investor.vanguard.com/investment-products/etfs/profile/bnd (accessed Oct. 04, 2022).

[49] "VXUS-Vanguard Total International Stock ETF | Vanguard." https://investor.vanguard.com/investment-products/etfs/profile/vxus (accessed Oct. 04, 2022).

[50] "Retirement Annuities | Annuity Solutions to Consider | Fidelity." https://www.fidelity.com/annuities/overview (accessed Oct. 04, 2022).

[51] "What is the Best Age to Buy An Annuity?," *Annuity.org*. https://www.annuity.org/annuities/buy/best-age-to-buy-an-annuity/ (accessed Oct. 03, 2022).

[52]"Alternatives to Annuities for Income in Retirement | What Are My Options?," *Annuity.org*. https://www.annuity.org/annuities/buy/alternatives/ (accessed Oct. 03, 2022).

[53] "Annuities | Investor.gov." https://www.investor.gov/introduction-investing/investing-basics/investment-products/insurance-products/annuities (accessed Oct. 04, 2022).

[54] "Annuity Income Calculator | Brighthouse Financial." [Online]. Available: https://www.brighthousefinancial.com/products/annuities/annuity-income-calculator/. [Accessed: Nov. 12, 2022]

[55] W. L. Winston and J. B. Goldberg, Operations Research: Applications and Algorithms, 4th ed. Belmont, CA: Thomson/Brooks/Cole, 2004.

[56]"How to Calculate Expected Portfolio Return," Investopedia. [Online]. Available: https://www.investopedia.com/ask/answers/061215/how-can-i-calculate-expected-return-my-portfolio.asp. [Accessed: Nov. 14, 2022]

[57] randerson112358, "Calculate Annualized Expected Stock Returns Using Python," Medium, Mar. 23, 2022. [Online]. Available: https://python.plainenglish.io/calculating-annualized-expected-stock-returns-using-python-aaba430ca8a9. [Accessed: Nov. 14, 2022]

[58] B. Brenyah, "Calculating Expected Rates of Returns for a Portfolio of stocks with Python," DS Biz, Sep. 18, 2017. [Online]. Available: https://medium.com/python-data/calculating-expected-rates-of-returns-for-a-portfolio-of-stocks-with-python-e3afbd4eeba5. [Accessed: Nov. 14, 2022]

[59]"What Is a Geometric Mean? How to Calculate and Example," Investopedia. [Online]. Available: https://www.investopedia.com/terms/g/geometricmean.asp. [Accessed: Nov. 14, 2022]

[60] "How to Calculate Your Investment Return," Investopedia. [Online]. Available: https://www.investopedia.com/articles/08/annualized-returns.asp. [Accessed: Nov. 14, 2022]

[61]"Volatility: Meaning In Finance and How it Works with Stocks," Investopedia. [Online]. Available: https://www.investopedia.com/terms/v/volatility.asp. [Accessed: Nov. 14, 2022]

[62] A. K. Srivastav, “Portfolio Standard Deviation,” WallStreetMojo, Oct. 10, 2018. [Online]. Available: https://www.wallstreetmojo.com/portfolio-standard-deviation/. [Accessed: Nov. 14, 2022]

[63] “5 Ways to Measure Mutual Fund Risk,” Investopedia. [Online]. Available: https://www.investopedia.com/investing/measure-mutual-fund-risk/. [Accessed: Nov. 14, 2022]

[64] “Beta: Definition, Calculation, and Explanation for Investors,” Investopedia. [Online]. Available: https://www.investopedia.com/terms/b/beta.asp. [Accessed: Nov. 15, 2022]

[65] “Portfolio Beta Calculator.” [Online]. Available: https://www.omnicalculator.com/finance/portfolio-beta. [Accessed: Nov. 15, 2022]

[66] “Sharpe Ratio Formula and Definition With Examples,” Investopedia. [Online]. Available: https://www.investopedia.com/terms/s/sharperatio.asp. [Accessed: Nov. 15, 2022]

[67]“Sharpe Ratio,” Corporate Finance Institute. [Online]. Available: https://corporatefinanceinstitute.com/resources/risk-management/sharpe-ratio-definition-formula/. [Accessed: Nov. 15, 2022]

[68] khanal, samir. “Flask-RESTful API With Heroku.” *Analytics Vidhya*, 19 Jan. 2021, https://medium.com/analytics-vidhya/flask-restful-api-with-heroku-da1ecf3e04b.

[69] “Annuity Calculator | Calculate Your Payout,” Annuity.org. [Online]. Available: https://www.annuity.org/annuities/immediate/annuity-calculator/. [Accessed: Nov. 16, 2022]

[70] “Investment Growth Retirement Calculator,” Personal Finance Club, Jul. 07, 2018. [Online]. Available: https://www.personalfinanceclub.com/investment-growth-retirement-calculator/. [Accessed: Nov. 16, 2022]

[71]“cFIREsim.” [Online]. Available: https://www.cfiresim.com/. [Accessed: Nov. 16, 2022]

[72]“Trinity study,” Wikipedia. Jan. 15, 2022 [Online]. Available: https://en.wikipedia.org/w/index.php?title=Trinity_study&oldid=1065774281. [Accessed: Nov. 16, 2022]

[73] What is asset allocation? Best asset allocation strategies! (n.d.). Napkin Finance. Retrieved November 16, 2022, from https://napkinfinance.com/napkin/what-is-asset-allocation/

[74] What are Annuities? (n.d.). Napkin Finance. Retrieved November 16, 2022, from https://napkinfinance.com/napkin/annuities/

[75] W3Schools online HTML editor. (n.d.). Retrieved November 16, 2022, from https://www.w3schools.com/howto/tryit.asp?filename=tryhow_css_flip_card

[76] Chart.js | Chart.js. (n.d.). Retrieved November 16, 2022, from https://www.chartjs.org/docs/latest/

[77] Chart.js. (n.d.). Retrieved November 16, 2022, from https://www.w3schools.com/js/js_graphics_chartjs.asp

[78] Bootstrap(Front-end framework). (2022). In Wikipedia. https://en.wikipedia.org/w/index.php?title=Bootstrap_(front-end_framework)&oldid=1125612151

[79] What is web hosting? - Web hosting explained - aws. (n.d.). Amazon Web Services, Inc. Retrieved November 16, 2022, from https://aws.amazon.com/what-is/web-hosting/

[80] CuriousChemE, “FuelOptGLPK.” [Online]. Available: https://github.com/CuriousChemE/FuelOptGLPK#readme

[81] Salesforce Developers. Introduction to Heroku. https://www.slideshare.net/developerforce/df13-introduction-to-heroku

[82]“Flask (Web Framework).” Wikipedia, 7 Nov. 2022. Wikipedia, https://en.wikipedia.org/w/index.php?title=Flask_(web_framework)&oldid=1120483147

[83]Debbie. “How to Deploy a Python + Flask API on Heroku.” *Medium*, 17 Jan. 2022, https://levelup.gitconnected.com/how-to-deploy-a-python-flask-api-on-heroku-2e5ddfd943ef.

[84] “Research Experience,” *Fidelity Research & Quotes*. [Online]. Available: https://digital.fidelity.com/prgw/digital/research/src. [Accessed: Nov. 10, 2022]

[85] “Fund Filter,” *Vanguard Investments* [Online]. Available: https://institutional.vanguard.com/fund-list/. [Accessed: Nov. 10, 2022]

[86] “Contribution Limits for IRAs, 401(k)s in 2022,” Investopedia. [Online]. Available: https://www.investopedia.com/retirement/now-you-can-save-more-retirement/. [Accessed: Nov. 10, 2022]

[87] “Retirement Topics — Required Minimum Distributions (RMDs) | Internal Revenue Service.” [Online]. Available: https://www.irs.gov/retirement-plans/plan-participant-employee/retirement-topics-required-minimum-distributions-rmds. [Accessed: Nov. 10, 2022]

[88] ‘What Is a Target-Date Fund? Risk Tolerance and Example’, *Investopedia*. [Online]. Available: https://www.investopedia.com/terms/t/target-date_fund.asp. [Accessed: Dec. 06, 2022]

[89] “Growing Your Investment,” ADP Financial Education powered by EVERFI. https://adp.everfi-next.net/student/dashboard/financialeducation/adp-calculators-resources/170?locale=en (accessed Sep. 23, 2022).

[90] “ 401k Contribution Calculator.” Fidelity. https://nb.fidelity.com/public/nb/401k/tools/calculators/contributioncalculator (accessed Sep. 23, 2022).

[91] “Asset Allocation Calculator,” Bankrate. https://www.bankrate.com/investing/asset-allocation-calculator/ (accessed Sep. 23, 2022).

[92] “401k Calculator - Bankrate.com,” Bankrate. https://www.bankrate.com/retirement/401-k-calculator/ (accessed Sep. 23, 2022).

[93] “What is a Fixed Annuity? - Nationwide.” https://www.nationwide.com/lc/resources/investing-and-retirement/articles/what-is-a-fixed-annuity (accessed Oct. 02, 2022).

[94] “Types of Annuities - Nationwide.” https://www.nationwide.com/lc/resources/investing-and-retirement/articles/best-annuities-for-you (accessed Oct. 02, 2022).

[95] “Income Annuity Estimator: Calculate Your Payout,” Schwab Brokerage. https://www.schwab.com/annuities/fixed-income-annuity-calculator (accessed Oct. 02, 2022).

[96] “Annuity Fees and Commissions | The Cost of Having an Annuity,” Annuity.org. https://www.annuity.org/annuities/fees-and-commissions/ (accessed Oct. 02, 2022).

[97]‘Vanguard - Investor Questionnaire - Terms and conditions of use’. [Online]. Available: https://retirementplans.vanguard.com/VGApp/pe/PubQuizActivity?FW_Event=IQViewEvt&ANTI_CSRF_TOKEN=4KHZ-JE73-5287-MU5Y-F6QZ-VJR2-2VUX. [Accessed: Dec. 06, 2022]

[98] contributors, M. O., Jacob Thornton, and Bootstrap. (n.d.). Get started with Bootstrap. Retrieved November 16, 2022, from https://getbootstrap.com/docs/5.2/getting-started/introduction/

[99] contributors, M. O., Jacob Thornton, and Bootstrap. (n.d.). Cards. Retrieved December 5, 2022, from https://getbootstrap.com/docs/5.2/components/card/

[100] contributors, M. O., Jacob Thornton, and Bootstrap. (n.d.). Modal. Retrieved December 5, 2022, from https://getbootstrap.com/docs/5.2/components/modal/

[101] contributors, M. O., Jacob Thornton, and Bootstrap. (n.d.). Tables. Retrieved December 5, 2022, from https://getbootstrap.com/docs/4.0/content/tables/

[102] Chart.js | Chart.js. (n.d.). Retrieved November 16, 2022, from https://www.chartjs.org/docs/latest/

[103] Line chart | chart. Js. (n.d.). Retrieved December 5, 2022, from https://www.chartjs.org/docs/3.9.1/charts/line.html

[104] uminder. (2022, April 9). Answer to “chart js using for loop.” Stack Overflow. https://stackoverflow.com/a/71806384

[105] LeeLenalee. (2021, November 3). Answer to “remove undefined label in the upper part of a chart.” Stack Overflow. https://stackoverflow.com/a/69825583

[106] Doughnut and pie charts | chart. Js. (n.d.). Retrieved December 5, 2022, from https://www.chartjs.org/docs/3.9.1/charts/doughnut.html#pie

[107] Candlestick charts. (n.d.). Google Developers. Retrieved December 5, 2022, from https://developers.google.com/chart/interactive/docs/gallery/candlestickchart

[108] Handling events | charts. (n.d.). Google Developers. Retrieved December 5, 2022, from https://developers.google.com/chart/interactive/docs/events

[109] Line chart | charts. (n.d.). Google Developers. Retrieved December 5, 2022, from https://developers.google.com/chart/interactive/docs/gallery/linechart

[110] Tooltips | charts. (n.d.). Google Developers. Retrieved December 5, 2022, from https://developers.google.com/chart/interactive/docs/customizing_tooltip_content

[111] Clayton, C. (2015, December 9). Answer to “How to Customize the tooltip being displayed in Google Vislization charts.” Stack Overflow. https://stackoverflow.com/a/34169017

[112] js.foundation, J. F.-. (n.d.). Slide effect | jquery ui api documentation. Retrieved December 5, 2022, from https://api.jqueryui.com/slide-effect/

[113] divanov11. (2019). Json-html-table. https://github.com/divanov11/json-html-table/blob/master/jsontable.html

[114] Swensen, M. (2021, May 10). Using svg vs. Canvas: A short guide. LogRocket Blog. https://blog.logrocket.com/svg-vs-canvas/

[115] Web design 101: How html, css, and javascript work. (n.d.). Retrieved December 6, 2022, from https://blog.hubspot.com/marketing/web-design-html-css-javascript

[116] Retirement free icons designed by surang. (n.d.). Flaticon. Retrieved December 6, 2022, from https://www.flaticon.com/free-icon/retirement_5572139

[117] Retirement Planning: An Introductory Guide to Retirement Plans in India." CreditMantri, https://www.creditmantri.com/article-retirement-planning-an-introductory-guide-to-retirement-plans-in-india/Accessed 6 Dec. 2022.

[118]"How to send JSON from JavaScript to Flask using Fetch API," Rahul baran's Blog. [Online]. Available: https://rahulbaran.hashnode.dev/how-to-send-json-from-javascript-to-flask-using-fetch-api. [Accessed: Dec. 06, 2022]

[119] "JavaScript, fetch, and JSON — Flask Documentation (2.2.x)." [Online]. Available: https://flask.palletsprojects.com/en/2.2.x/patterns/javascript/. [Accessed: Dec. 06, 2022]

[120]Learn Flask for Python - Full Tutorial. [Online]. Available: https://www.youtube.com/watch?v=Z1RJmh_OqeA. [Accessed: Dec. 06, 2022]

[121]“How to get the value of a form element using JavaScript | JavaScript Coder.” [Online]. Available: https://www.javascript-coder.com/javascript-form/javascript-get-form/. [Accessed: Dec. 06, 2022]

[122]“Flask HTTP methods, handle GET & POST requests - Python Tutorial.” [Online]. Available: https://pythonbasics.org/flask-http-methods/. [Accessed: Dec. 06, 2022]

[123]”Sending Form Data (POST) with the Fetch API in JavaScript.” [Online]. Available: https://www.youtube.com/watch?v=c3qWHnJJbSY. [Accessed: Dec. 06, 2022]

[124]“Response.text() - Web APIs | MDN.” [Online]. Available: https://developer.mozilla.org/en-US/docs/Web/API/Response/text. [Accessed: Dec. 06, 2022]

[125]“HTMLFormElement: submit event - Web APIs | MDN.” [Online]. Available: https://developer.mozilla.org/en-US/docs/Web/API/HTMLFormElement/submit_event . [Accessed: Dec. 06, 2022]

[126] “Using the Fetch API - Web APIs | MDN.” [Online]. Available: https://developer.mozilla.org/en-US/docs/Web/API/Fetch_API/Using_Fetch. [Accessed: Dec. 06, 2022]

[127]“HTML onclick Event Attribute.” [Online]. Available: https://www.w3schools.com/tags/ev_onclick.asp. [Accessed: Dec. 06, 2022]

[128]HoldOffHunger, “Answer to ‘Waiting for the fetch to complete and then execute the next instruction,’” Stack Overflow, Aug. 23, 2018. [Online]. Available: https://stackoverflow.com/a/51992739. [Accessed: Dec. 06, 2022]

[129]“async function - JavaScript | MDN.” [Online]. Available: https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Statements/async_function. [Accessed: Dec. 07, 2022]

[131]“How to Calculate Annualized Portfolio Return: 10 Steps,” wikiHow. [Online]. Available: https://www.wikihow.com/Calculate-Annualized-Portfolio-Return. [Accessed: Dec. 07, 2022]

[132] "Annual Return," Corporate Finance Institute. [Online]. Available: https://corporatefinanceinstitute.com/resources/wealth-management/annual-return/. [Accessed: Dec. 07, 2022]

[133] "2- Using Python & Pandas Datareader to Analyze Financial Data." [Online]. Available: https://www.datasource.ai/uploads/1c79c40efac8fff409e7c23fe8167d04.html. [Accessed: Dec. 07, 2022]

[134]"Standard Deviation Indicator - Fidelity." [Online]. Available: https://www.fidelity.com/learning-center/trading-investing/technical-analysis/technical-indicator-guide/standard-deviation. [Accessed: Dec. 07, 2022]

[135]"What Is SMA? - Simple Moving Average - Fidelity." [Online]. Available: https://www.fidelity.com/learning-center/trading-investing/technical-analysis/technical-indicator-guide/sma. [Accessed: Dec. 07, 2022]

[136]"4 Steps to Building a Profitable Portfolio," Investopedia. [Online]. Available: https://www.investopedia.com/financial-advisor/steps-building-profitable-portfolio/. [Accessed: Dec. 07, 2022]

[137] "How Financial Advisors Pick Client Investments," Investopedia. [Online]. Available: https://www.investopedia.com/articles/financial-advisors/052115/how-financial-advisors-pick-client-investments.asp. [Accessed: Dec. 07, 2022]

[138]W. Jeffries and A. Brodsky, "Composite Alternative Pareto Optimal Recommendation System with Individual Utility Extraction (CAPORS-IUX):," in Proceedings of the 20th International Conference on Enterprise Information Systems, Funchal, Madeira, Portugal, 2018, pp. 328–335, doi: 10.5220/0006770403280335 [Online]. Available: http://www.scitepress.org/DigitalLibrary/Link.aspx?doi=10.5220/0006770403280335. [Accessed: Dec. 07, 2022]

[139]"Best Annuity Rates of 2022," Investopedia. [Online]. Available: https://www.investopedia.com/best-annuity-rates-5179335. [Accessed: Dec. 07, 2022]

[140] "pandas.Series.size — pandas 1.5.2 documentation." [Online]. Available: https://pandas.pydata.org/docs/reference/api/pandas.Series.size.html. [Accessed: Dec. 07, 2022]

[141]"pandas.Series.iloc — pandas 1.5.2 documentation." [Online]. Available: https://pandas.pydata.org/docs/reference/api/pandas.Series.iloc.html. [Accessed: Dec. 07, 2022]

[142]"Python math.pow() Method." [Online]. Available: https://www.w3schools.com/python/ref_math_pow.asp. [Accessed: Dec. 07, 2022]

[143]“Pandas - Data Correlations.” [Online]. Available: https://www.w3schools.com/python/pandas/pandas_correlations.asp. [Accessed: Dec. 07, 2022]

[144] “Pandas - Cleaning Empty Cells.” [Online]. Available: https://www.w3schools.com/python/pandas/pandas_cleaning_empty_cells.asp. [Accessed: Dec. 07, 2022]

[145]“pandas.DataFrame — pandas 1.5.2 documentation.” [Online]. Available: https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.html. [Accessed: Dec. 07, 2022]

[146]“Indexing and selecting data — pandas 1.5.2 documentation.” [Online]. Available: https://pandas.pydata.org/docs/user_guide/indexing.html. [Accessed: Dec. 07, 2022]

[147]B. Nicholson, “Answer to ‘Finding Pyomo provided math functions,’” Stack Overflow, Jan. 10, 2018. [Online]. Available: https://stackoverflow.com/a/48192897. [Accessed: Dec. 07, 2022]

[148] “EC2 - Get Started,” Amazon Web Services, Inc. [Online]. Available: https://aws.amazon.com/ec2/ec2-get-started/. [Accessed: Dec. 07, 2022]

[149] ‘Publication 590-B (2021), Distributions from Individual Retirement Arrangements (IRAs) | Internal Revenue Service’. [Online]. Available: https://www.irs.gov/publications/p590b. [Accessed: Dec. 07, 2022]

[150] “Retirement plans in the USA,” The American Dream. [Online]. Available: https://www.the-american-dream.com/retirement-plan-usa/. [Accessed: Dec. 07, 2022]

[151] “What is an IRA? | why you should invest | Fidelity.” [Online]. Available: https://www.fidelity.com/building-savings/learn-about-iras/what-is-an-ira. [Accessed: Dec. 07, 2022]

[152] “Health Savings Account (HSA): How HSAs Work, Contribution Rules,” Investopedia. [Online]. Available: https://www.investopedia.com/terms/h/hsa.asp. [Accessed: Dec. 07, 2022]

[153] “Health Savings Account (HSA) - Glossary,” HealthCare.gov. [Online]. Available: https://www.healthcare.gov/glossary/health-savings-account-hsa. [Accessed: Dec. 07, 2022]

[154] “What Is A Brokerage Account? – Forbes Advisor.” [Online]. Available: https://www.forbes.com/advisor/investing/what-is-a-brokerage-account/. [Accessed: Dec. 07, 2022]

[155] “ETF vs. Mutual Fund: It Depends on Your Strategy,” Schwab Brokerage. [Online]. Available:

https://www.schwab.com/learn/story/etf-vs-mutual-fund-it-depends-on-your-strategy. [Accessed: Dec. 07, 2022]

[156] “Mutual Fund vs. ETF: What’s the Difference?,” Investopedia. [Online]. Available: https://www.investopedia.com/articles/exchangetradedfunds/08/etf-mutual-fund-difference.asp. [Accessed: Dec. 07, 2022]

[157]“How to Reduce Volatility in Your Portfolio,” Investopedia. [Online]. Available: https://www.investopedia.com/financial-edge/0312/how-to-reduce-volatility-in-your-portfolio.aspx. [Accessed: Dec. 07, 2022]

[158]‘Typical Mutual Fund Expense Ratios’, NerdWallet. [Online]. Available: https://www.nerdwallet.com/article/investing/mutual-fund-expense-ratios. [Accessed: Dec. 07, 2022]

[159]A. Brodsky and J. Luo, “Decision Guidance Analytics Language (DGAL) - Toward Reusable Knowledge Base Centric Modeling:,” in Proceedings of the 17th International Conference on Enterprise Information Systems, Barcelona, Spain, 2015, pp. 67–78, doi: 10.5220/0005349600670078 [Online]. Available: http://www.scitepress.org/DigitalLibrary/Link.aspx?doi=10.5220/0005349600670078. [Accessed: Dec. 07, 2022]

Appendix A

Please look at our code’s comments, which lists links (included here and/or in the code file) referenced to form specific parts of the code.

<u>Backend Sources:</u>

We referenced many sources to construct our code for backend logic. Specifically, to understand steps to build a Flask server we mainly referenced, [68], [118]-[120]. To debug data transmission between our HTML forms and Flask server, we referenced [121]-[127]. Additionally, we followed steps from [80] and [68] to deploy our Flask server code to Heroku. To format fetch calls (containing POST, and GET requests) to our APIs in our front end code (script.js), we referenced/used the syntax from [126]. Additionally, we reused code structure from [128] and [129] to create asynchronous fetch calls (which wait for our server to send back data). As for our backend, the references we used to compute each metric and constraint are listed in Table 2 Formula/Description columns. Additional sources we used to get information on specific syntax used in our backend code include sources from the Pandas and W3 School Documentation ([140]- [146]),and to address issues with Pyomo ([147])

<u>Front End Sources on Tools used/Sources:</u>

FLASK:

- Flask is a micro web application framework written in Python and based on
  - WSGI toolkit and
  - Jinja template engine

- Flask is extensible while keeping the core simple. Examples: Flask-SQLAlchemy, Flask-Mail, Flask-Admin, Flask-Cache
- The applications that make use of the Flask framework are Pinterest, LinkedIn as well as the community web page for Flask itself.

HEROKU:

- Heroku is an elastic multi-language, multi-framework platform as a service.
- Heroku is built on a managed container system and runs apps in smart containers called DINOS.
- Developers can deploy their application's code in any of the languages Heroku supports as dinos work in a fully managed runtime environment.
- Its supported languages include nodeJS, ruby, Java, PHP, python, go, scala, and closure, or you can use another language through a third-party build pack as long as it runs on Linux.
- Heroku takes care of monitoring and patching the system and language stacks to keep the platform up-to-date, reliable and secure.

Steps to deploy Python Web API to Heroku:

- Before deploying we need to generate two files in the root folder.
  - Procfile
  - requirements.txt
- Install a python server called gunicorn → pip install gunicorn
- After installing, generate a new file called Procfile → echo> Procfile
- Add this line web: gunicorn main : app in Procfile
- Now we will generate a requirements file by using the command
  → pip freeze>requirements.txt
- Now, make use of GitHub to deploy the code to Heroku by linking your GitHub account to Heroku.
- GLPK was installed via the use of an Aptfile and Heroku's build packs for python applications ([80] explains this is needed for files not in Python package).

- Followed the steps (from [80]) below to create the Heroku app using GIT
  ```
  $ git init
  $ git add .
  $ git commit -m "first commit"
  $ heroku create cs787-proj-app
  $ heroku buildpacks:set heroku/python
  $ heroku buildpacks:add --index 1 heroku-community/apt
  $ git push heroku master
  ```

HTML:

- HTML stands for HyperText Markup Language. "Markup language" means that, rather than using a programming language to perform functions.
- It uses tags to identify different types of content and the purposes they each serve to the webpage and provides the basic structure of sites, which is enhanced and modified by other technologies like CSS and JavaScript.
- Using HTML, we can add headings, format paragraphs, control line breaks, make lists, emphasize text, create special characters, insert images, create links, build tables, control some styling, and much more.

CSS:

HTML provides the raw tools needed to structure content on a website. CSS (Cascading Style Sheets), on the other hand, helps to style this content so it appears to the user the way it was intended to be seen.

JavaScript:

- JavaScript is a logic-based programming language that can be used to modify website content and make it behave in different ways in response to a user's actions.
- Common uses for JavaScript include confirmation boxes, calls-to-action, and adding new identities to existing information.
- In short, JavaScript is a programming language that lets web developers design interactive sites.

Bootstrap (v5.2.3):

Bootstrap is a free, open source CSS framework to help design websites faster and easier. We used the following bootstrap's HTML and CSS based design templates -

- **Cards**: A card is a flexible and extensible content container. It includes options for headers and footers, a wide variety of content, contextual background colors, and powerful display options.
- **Modal**: modal plugins are used to add dialogs to the site for lightboxes, user notifications, or completely custom content. Modals are built with HTML, CSS, and JavaScript. They're positioned over everything else in the document and remove scroll from the <body> so that modal content scrolls instead. Clicking on the modal "backdrop" will automatically close the modal.
- **Table**
- **Tooltips**

Chart JS (v3.9.1):

- Chart.js is a free, open-source JavaScript library for data visualization and it provides a selection of ready to go charts which can be styled and configured.
- It supports Scatter, Line, Bar, Pie, Donut, Bubble, Area, Radar, Mixed chart types.
- It renders chart elements on an HTML5 canvas unlike several other charting libraries that render as SVG making it very performant, especially for large datasets.
- Line and Pie charts are being used in our project to visualize growth of balance in accounts over age and percentage(%) contribution to each account.

Google Charts:

- Google Charts provides a perfect way to visualize data on your website. From simple line charts to complex hierarchical tree maps, the chart gallery provides a large number of ready-to-use chart types.
- The most common way to use Google Charts is with simple JavaScript that is embedded in the web page.
  - load some Google Chart libraries,
  - list the data to be charted,
  - select options to customize your chart, and
  - finally create a chart object with an id.
  - then, later in the web page, create a <div> with that id to display the Google Chart.
- Charts are rendered using HTML5/SVG technology to provide cross-browser compatibility.
- We are using a Candlestick chart to display the confidence interval to the user.

EC2:

- We are using AWS EC2 for cloud hosting of our 'Retirement Planning Tool'.
- It allows us to obtain and configure virtual compute capacity in the cloud.
- EC2 also ensures support for security, ports, processors and networking facilities.