Phase 1: Planning

1. Identify the purpose and goals of the web application.
2. Create a list of requirements and features that the application must have.
3. Determine the target audience and user personas.
4. Create a project timeline with milestones and deliverables.

Phase 2: Design

1. Create wireframes and mockups of the application's user interface.
2. Design the database schema and create entity relationship diagrams (ERDs).
3. Choose a web framework (e.g. Flask, Django) and design the application architecture.
4. Plan the deployment strategy and choose a hosting provider (e.g. AWS, Heroku).

Phase 3: Development

1. Set up the development environment and project structure.
2. Create the application's models and database tables.
3. Implement the application's business logic and API endpoints.
4. Build the application's user interface and front-end functionality.
5. Add user authentication and authorization (if required).
6. Implement unit tests and integration tests to ensure quality and correctness.

Phase 4: Deployment and Testing

1. Deploy the application to the chosen hosting provider.
2. Conduct thorough testing to ensure the application is working correctly.
3. Monitor and optimize application performance and security.
4. Debug any issues and make necessary updates.

Phase 5: Maintenance and Support

1. Provide ongoing maintenance and support for the application.
2. Collect user feedback and implement necessary updates.
3. Address any security issues or bugs that arise.
4. Plan for future updates and feature enhancements.

# Phase 1: Planning

The purpose of this section is to define the plan for the project. The use case for this app is a place to track all my long-term assets and provide a summary of all assets across all asset classes.

## Purpose and goals

* Provides a summary of all assets across all asset classes. Drilldown capabilities.
* Stores information within a DB.
* Fetches live prices to give current value of the portfolio.
* Allows users to add stocks, crypto and funds.
* Allows users to add assets manually
* Allows users to bulk add using an excel sheet

## Timeline

# Phase 2: Design

## ERD

+-------------+

| Portfolio |

+-------------+

| id |

| name |

+-------------+

|

|

+----------+------------+

| |

| |

+-------------+ +-------------+

| AssetType | | Asset |

+-------------+ +-------------+

| id | | id |

| name | | name |

+-------------+ | symbol |

| | asset\_type |

| +-------------+

|

+-------------------+

| |

| |

+---------------+ +---------------+ +--------------+

| Stock | | Crypto | | Fund |

+---------------+ +---------------+ +--------------+

| id | | id | | id |

| symbol | | symbol | | name |

| name | | name | | ticker |

| price | | price | | asset\_id |

| asset\_id | | asset\_id | | asset\_type |

+---------------+ +---------------+ +--------------+

## Web framework

### Options

* + Flask
  + Django
  + Jinja2

### Infrastructure

## Deployment