

EduConnect Project Documentation

1. Project Overview

EduConnect is an educational opportunity management system designed to help users (admins) add, search, and manage various educational opportunities such as internships, workshops, lectures, and volunteer opportunities. It allows admins to store, retrieve, and display opportunities from a file, making it easier for users to access relevant information for educational purposes.

2. Technologies Used

- C++: The core programming language for the project.
- STL (Standard Template Library): Used for storing and managing collections of opportunities (via `std::vector`).
- File I/O: For saving and loading data from text files (`opportunities.txt`).

3. Features

- Add Opportunity: Admins can add new educational opportunities by providing relevant details (e.g., title, description, category, time, location).
- Search Opportunities: Admins can search for opportunities based on various criteria, such as title, location, or category.
- File Persistence: Opportunities are saved to a file and can be loaded when the program starts, ensuring data persistence between sessions.
- Dynamic User Interaction: Admins can interact with the system through a terminal interface, using simple input prompts to manage opportunities.

4. Class Descriptions

DatabaseManager: The DatabaseManager class is responsible for managing the collection of opportunities. It provides functionality to:

- Add opportunities.
- Display all stored opportunities.
- Save and load opportunities to/from a file.
- Retrieve the total count of stored opportunities.

Admin: The Admin class acts as the interface for adding and searching educational opportunities. It includes methods for:

- Adding new opportunities to the database.
- Searching for opportunities by title, location, or category.

SearchSystem: The SearchSystem class enables the searching of stored opportunities. It provides methods to:

- Search opportunities by title.
- Search opportunities by location.
- Search opportunities by category.

Opportunity: The Opportunity class represents an individual educational opportunity. It stores details such as:

- Title, description, category, time, and location.
It also provides methods to display the opportunity's details and retrieve individual properties via getters.

FileManager: The FileManager class handles file I/O operations for saving and loading opportunities from a file. It provides static methods to:

- Save opportunities to a file.
- Load opportunities from a file.

5. Method Details

Admin Methods

- `addOpportunityToDatabase()`:
Prompts the admin to input details for a new opportunity and adds it to the DatabaseManager. Calls the `saveOpportunitiesToFile()` method to persist the new opportunity.
- `searchOpportunity()`:
Displays a menu with options to search by title, location, or category. Based on the user's choice, it calls the corresponding search method in SearchSystem.

SearchSystem Methods

- `searchOpportunity()`:
Displays a menu for the user to select search criteria. Invokes `searchByTitle()`, `searchByLocation()`, or `searchByCategory()` based on the user's input.

`searchByTitle()`, `searchByLocation()`, `searchByCategory()`:

Each method loops through the opportunities in the DatabaseManager and compares the selected attribute (title, location, or category) with the stored opportunities.

DatabaseManager Methods

- `addOpportunity()`:
 - Adds a new Opportunity object to the internal collection.
- `displayAllOpportunities()`:
 - Iterates over the stored opportunities and calls `displayOpportunityInfo()` to print each opportunity's details to the console.
- `saveOpportunitiesToFile()`:
 - Writes all stored opportunities to a specified file (`opportunities.txt`) in CSV format.
- `loadOpportunitiesFromFile()`:
 - Loads opportunities from a specified file and adds them to the database.
- `getOpportunityCount()`:
 - Returns the number of opportunities stored in the database.
- `getOpportunity()`:
 - Returns a pointer to an opportunity at the specified index.

FileManager Methods

- saveOpportunitiesToFile():
 - Saves all opportunities from the DatabaseManager to a file in CSV format.
- loadOpportunitiesFromFile():
 - Loads opportunities from a file and adds them to the DatabaseManager.

Conclusion:

The EduConnect project provides an easy-to-use system for managing and searching educational opportunities. With features like file persistence, user-friendly searching, and the ability to add new opportunities, the system ensures efficient handling of educational data.