Good Reads

Problem:

Your aim is to build a console application for a books platform where users can register, purchase and read books.
You need to provide following options:

- AddBook: Add a book to the collection of books. This operation can only be done by the admin user (which is a part of your application and is not needed to be registered).
- Register, Login and LogOut: A user can sign up and login (No need for Authentication, name can be considered as id of the user). Login makes users active and logout makes them inactive.
- Purchase: Users can purchase a book post login.
- StartRead/ResumeRead: Users can start reading a book and can resume from where they left off (both start and resume functions the same way).
- Next: Once a user is reading a book, he/she can turn to the next page.
- Previous: Once a user is reading a book, he/she can turn to the previous page.
- Rate: Users can rate a book on a scale of 1 to 10.
- GoTo: Users can jump to a page in the book he/she is reading.
- ListMyBooks: List all of the users' books.

Data:
We can consider a english paragraph (without any new line and special characters) as a book and it's words as pages (considering . and , as part of the word/page).

Bonus:

- ListBooks: List all books in the platform library and their ratings.
- RecommendBook: Recommend a book to a user.
  - Recommend at max 5 books to the user, sorted on their ratings (highest rating first). Books with no ratings are not considered for recommendation.
  - Recommendations should not have books the user has already read or is reading but the books user has purchased but has not started reading can appear in recommendations.
- Accept page size as an input (default will be one word) and complete words upto this limit should be displayed on the screen.
  - E.g. for a page size of 10 characters, pages will look like:
    - `A woman`
    - `finds a`
    - `pot of`
    - `treasure`

Input and Output (You may change the input output format without changing the functionality to suit your needs):

```
                                     → is input
                                     ← is output
→ AddBook
← Enter the title and content of the book.
→ The Bogey Beast
→ A woman finds a pot of treasure on the road while she is returning from work.
Delighted with her luck, she decides to keep it. As she is taking it home, it keeps
changing. However, her enthusiasm refuses to fade away.
← Book added successfully.
→ Login tej
← Welcome tej
→ Purchase The Bogey Beast
← Book added to your library.
→ StartRead The Bogey Beast
← A
→ Next
← woman
→ GoTo 10
← road
```

Guidelines:

- Input can be read from a file or STDIN or coded in a driver method.
- Output can be written to a file or STDOUT.
- Store all interim/output data in-memory. Usage of databases is not allowed.
- Restrict internet usage to looking up syntax and for english paragraphs only (you can go to this link for short stories https://www.fluentu.com/blog/english/easy-english-short-stories-2/).
- You are free to use the language of your choice.
- Save your code/project by your name and email it. Your program will be executed on another machine. So, explicitly specify dependencies, if any, in your email.

Expectations:

- Code should be demo-able (very important). Code should be functionally correct and complete.
  - At the end of this interview round, an interviewer will provide multiple inputs to your program for which it is expected to work
- Code should handle edge cases properly and fail gracefully. Add suitable exception handling, wherever applicable.
  - An example would be to display an error message saying “You aren’t reading a book.” when the user is trying to use the GoTo command without the Resume/Start read command before it.
- Code should have good object-oriented design.
- Code should be readable, modular, testable and extensible. Use intuitive names for your variables, methods and classes.
  - It should be easy to add/remove functionality without rewriting a lot of code.
  - Do not write monolithic code.
- Don’t use any databases.