Problem: Library Management System

You are tasked with building a Library Management System that allows the management of various types of users and books. The system should allow operations like borrowing, returning, and displaying book details based on the type of user.

Requirements:

1. Users:

There are two types of users in the system:

- **Student**: Can borrow up to 3 books at a time.
- **Faculty**: Can borrow up to 10 books at a time.

Create a common **User** interface that defines the following methods:

- **borrowBook(Book book)**: Handles borrowing a book.
- **returnBook(Book book)**: Handles returning a book.
- **getBorrowedBooks()**: Returns a list of currently borrowed books.

Both **Student** and **Faculty** should implement this interface, with specific borrowing limits for each.

2. Books:

There are two types of books in the library:

- **PrintedBook**: Requires a physical copy to be borrowed.
- **EBook**: Can be borrowed digitally without limits on the library stock.

Create an abstract **Book** class that defines the common properties of a book:

- **title**: The title of the book.
- **author**: The author of the book.
- **isAvailable**: (For physical books only) Tracks whether the book is available or already borrowed.

Include an abstract method:

- **getDetails()**: Prints details of the book.

Implement two subclasses: **PrintedBook** and **EBook**, each with specific details and behaviors.

3. Library Operations:

- Use **polymorphism** to allow both **Student** and **Faculty** users to borrow either type of book through the same interface.
- If a **Student** tries to borrow more than 3 books or a **Faculty** tries to borrow more than 10 books, the system should reject the request with an appropriate message.
- For **PrintedBook**, check availability before allowing borrowing. For **EBook**, no availability check is needed.

4. Display Book Details:

- Use a method **displayBookDetails(Book book)** to display book information, taking advantage of polymorphism to show specific details for either **PrintedBook** or **EBook**.

5. Implementation Notes:

- Demonstrate the use of inheritance for **PrintedBook** and **EBook**.
- Use interfaces for user behavior.
- Use polymorphism to handle different user types and book types seamlessly.

Submission Format

1. The name of main class should be Roll_X (X denoting your roll).
2. submit your java file it in google classroom.
3. Use-proper commentation and indentation.
4. Before submission, show your solution to your lab teacher.