**OOP Exercise-1 : Job Portal Console Application**

**Objective:**

Create a console application for a job portal system called "HireMeNow." The application should allow job seekers to register, log in, view jobs, save jobs, apply for jobs, and manage their profiles.

**Requirements:**

**1. Job Class (`models.Job`):**

- Fields:

- `Id` (int): Unique identifier for the job.

- `Title` (string): Title of the job.

- `ExperienceLevel` (ExperienceLevels enum): Experience level required for the job (Fresher, MidLevel, Senior).

- `Company` (string): Name of the hiring company.

- `Location` (string): Location of the job.

- `SalaryRange` (string): Range of salary for the job.

- `JobType` (string): Type of the job.

**2. JobSeeker Class (`models.JobSeeker`):**

- Fields:

- `Id` (int): Unique identifier for the job seeker.

- `FirstName` (string): First name of the job seeker.

- `LastName` (string): Last name of the job seeker.

- `Email` (string): Email address of the job seeker.

- `Phone` (string): Phone number of the job seeker.

- `Location` (string): Location of the job seeker.

- `AboutMe` (string): Brief description of the job seeker.

- `Qualification` (string): Educational qualification of the job seeker.

- `ExperienceLevel` (ExperienceLevels enum): Experience level of the job seeker.

- `Password` (string): Password for job seeker authentication.

- Methods:

- `addAppliedJob(Job job)`: Add a job to the list of applied jobs.

- `GetAppliedJobs()`: Retrieve the list of applied jobs.

- `addSavedJob(Job job)`: Add a job to the list of saved jobs.

- `GetSavedJobs()`: Retrieve the list of saved jobs.

**3. JobManager Class (`managers.JobManager`):**

- Fields:

- `Jobs` (Job[]): An array to store available jobs.

- Methods:

- `ListJobs()`: Display a list of all available jobs.

- `PrintJobs(Job[] jobs)`: Print details of the provided array of jobs.

- `GetJobById(int jobId)`: Retrieve a job by its unique identifier.

**4. JobSeekerManager Class (`managers.JobSeekerManager`):**

- Fields:

- `jobSeekers` (JobSeeker[]): An array to store registered job seekers.

- `loggedInJobSeeker` (JobSeeker): The currently logged-in job seeker.

- `jobManager` (JobManager): An instance of `JobManager` to handle job-related operations.

- Methods:

- `RegisterJobSeeker()`: Register a new job seeker and add them to the `jobSeekers` array.

- `LoginJobSeeker()`: Log in an existing job seeker and set `loggedInJobSeeker` if successful.

- `ShowJobSeekerMenu()`: Display the main menu for logged-in job seekers.

- `ApplyJob()`: Allow the job seeker to apply for a job.

- `SaveJob()`: Allow the job seeker to save a job.

- `ViewProfile()`: Display the profile details of the logged-in job seeker.

- `Logout()`: Log out the currently logged-in job seeker.

- `ShowMainMenu()`: Display the main menu for the application.

**5. ExperienceLevels Enum (`enums.ExperienceLevels`):**

- Enum values: `Fresher`, `MidLevel`, `Senior`.

Task Assignment:

Your task is to create a console application based on the provided code snippets and specifications. Ensure that the application meets the requirements outlined above. Pay attention to details and follow best practices in C# programming.

**Note:**

- You can test the application by running it and performing various actions as a job seeker.

- Make sure to handle edge cases and provide appropriate feedback to users.

- Consider adding error handling and validation where necessary.

- Ensure that the code is well-organized and follows a clear structure.

**Submission:**

Share the completed C# console application code, and provide a brief overview of how you tested the application. Highlight any challenges faced during the implementation and how you addressed them.