JVM Assignment-2

```
com.byteminds.bytecraftpro
│
├── authentication
│   ├── LoginService.java
│   ├── RegistrationService.java
│   ├── User.java
│   └── PasswordUtils.java
│
├── content
│   ├── ContentManager.java
│   ├── DraftService.java
│   ├── PublishedContent.java
│   └── TaggingService.java
│
├── payment
│   ├── PaymentProcessor.java
│   ├── SubscriptionPlan.java
│   └── Invoice.java
│
├── util
│   ├── Logger.java
│   ├── ConfigReader.java
│   └── Validator.java
```

```
│

└── main

    ├── Application.java

    └── StartupRunner.java
```

Rationale for Package Organization

1. Modularity and Separation of Concerns

- **authentication**: Handles user authentication, login, registration, and password management, ensuring a clear focus on security and user access.
- **content**: Manages content-related operations such as drafting, publishing, and tagging, aligning with the primary functionality of ByteCraft Pro.
- **payment**: Centralizes payment processing, subscription plans, and invoice generation, ensuring clear separation from other business logic.
- **util**: Contains utility classes like logging and configuration readers, enabling code reuse across different modules.
- **main**: Houses the main application entry point and startup logic to bootstrap the project.

2. Maintainability

- A structured package hierarchy makes it easier for developers to locate specific classes.
- Modular packages simplify debugging, testing, and scaling individual components.

3. Scalability

- Additional functionality, like analytics or reporting, can be added as separate packages (e.g., `com.byteminds.bytecraftpro.analytics`).
- Classes can grow within their respective packages without disrupting the overall structure.

4. Reusability

- Utility classes in the `util` package can be reused across the project, promoting consistency and reducing duplication.

5. Industry Standards

The hierarchical naming convention (`com.byteminds.bytecraftpro`) follows Java best practices, ensuring:

- Unique package names (preventing conflicts with third-party libraries).
- Readability and alignment with industry norms.

Conclusion

This package organization ensures a modular, scalable, and maintainable codebase for ByteCraft Pro. Each module aligns with its functionality, adhering to Java best practices, and facilitates easy onboarding of new developers and future enhancements.