

Core Functionalities Implemented

User Authentication (Login & Register)

- JWT-based authentication with secure password hashing
- Singleton-based AuthService for consistent login/register logic
- Protected routes using middleware (authMiddleware.js)

Resume Analysis

- File upload support for .pdf and .docx
- Resume content extraction using pdf-parse and mammoth
- Resume analysis based on subscription:
 - Free users → KeywordResumeAnalyzer
 - Premium users → AIResumeAnalyzer

Role-Based User Types

- User model with role discriminator:
 - Student, FreshGraduate, Mentor
- Mongoose inheritance using discriminator

Notification System

- Observer pattern-based service to notify users after actions
 - Supports:
 - Email notifications
 - SMS notifications
 - Push notifications (simulated)
-

Key Design Patterns Implemented

1. Factory Pattern

📁 factories/ResumeAnalyzerFactory.js

Creates analyzer instances based on user subscription.

```
class ResumeAnalyzerFactory {
  static getAnalyzer(subscriptionType) {
    if (subscriptionType === 'premium') return new
AIResumeAnalyzer();
    return new KeywordResumeAnalyzer();
  }
}
```

2. Observer Pattern

📁 observers/

Implements dynamic notification system:

- NotificationService (Subject)
- Observer (Interface)
- EmailNotifier, SMSNotifier, PushNotifier (Concrete observers)

```
const notifier = new NotificationService();
notifier.attach(new EmailNotifier());
notifier.attach(new SMSNotifier());
notifier.notify("Resume analyzed!");
```

3. Singleton Pattern

📁 services/AuthService.js

Ensures a single instance of AuthService handles all auth logic.

```
class AuthService {
  static instance;
  constructor() {
    if (AuthService.instance) return AuthService.instance;
    AuthService.instance = this;
  }
  // register() and login() methods...
}
```

Folder Structure Summary

skillsync-backend/

```
|— models/
|   |— User.js
|   |— Student.js
|   |— FreshGraduate.js
|   |— Mentor.js
|   |— Observer.js
|   └─ Resume.js
|
|— controllers/
|   |— userController.js
|   |— authController.js
|   └─ resumeController.js
|
|— routes/
|   |— userRoutes.js
|   |— authRoutes.js
|   └─ resumeRoutes.js
|
|— services/
|   |— AuthService.js          # Singleton login/register logic
|   └─ analyzers/
|       |— ResumeAnalyzer.js  # Abstract analyzer
|       |— KeywordResumeAnalyzer.js
|       └─ AIResumeAnalyzer.js
|
|— factories/
|   └─ ResumeAnalyzerFactory.js
|
|— observers/
|   |— Observer.js            # Interface
|   |— EmailNotifier.js
|   |— SMSNotifier.js
|   |— PushNotifier.js
|   └─ NotificationService.js # Subject
|
|— middleware/
|   └─ authMiddleware.js      # JWT token validation
|
|— config/                    # (optional)
|   └─ db.js                  # DB connection (if extracted from
app.js)
```

```
|
|— utils/                                # (optional)
|   └─ tokenHelper.js                  # (if using custom helpers)
|
|— app.js                               # Express app setup + routing
|— server.js                           # Starts server
|— .env                                # Environment variables
|— package.json                        # Node.js dependencies
```

Tools & Libraries

Tool/Lib	Purpose
Node.js	Backend runtime
Express.js	Routing, middleware
MongoDB	Database
Mongoose	Schema modeling
jsonwebtoken	Auth token handling
bcrypt	Password hashing
multer	File upload
pdf-parse	PDF resume parsing
mammoth	DOCX resume parsing
dotenv	Load .env variables