.Net Assignment

Employee Performance Management System (EPMS) - Backend API Specification

1. Authentication & Authorization

- Endpoints

1.1 User Login

`POST /api/auth/login`

Request:

```Json
{
  "email": "user@example.com",
  "password": "SecurePass123!"
}
```

Response:

```Json
{
  "token": "eyJhbGciOiJIUz...",
  "expiresIn": 3600,
  "user": {
    "id": 1,
    "name": "John Doe",
    "role": "HR"
  }
}
```

---

2. User & Role Management

◆ Endpoints

2.1 Create User (HR Only)

```
POST /api/users
```

Request:

```json
{
  "name": "Jane Doe",
  "email": "jane.doe@example.com",
  "password": "StrongPass@123",
  "role": "Manager"
}
```

Response:

```json
{
  "id": 2,
  "name": "Jane Doe",
  "email": "jane.doe@example.com",
  "role": "Manager",
  "createdAt": "2024-02-05T12:00:00Z"
}
```

Constraints:

☑ **Only HR** can create users.

☑ Email must be **unique**.

☑ Passwords must have **min 8 chars, 1 uppercase, 1 special char**.

2.2 Get Users (HR Only)

```
GET /api/users?role=Employee&page=1&limit=10
```

Response:

```json
{
  "data": [
    {
      "id": 1,
      "name": "John Doe",
      "email": "john.doe@example.com",
      "role": "Employee"
```

```
    }
  ],
  "pagination": {
    "currentPage": 1,
    "totalPages": 3,
    "totalItems": 25
  }
}

```

Constraints:

✅ Only **HR** can fetch users.

✅ Supports **pagination & filtering** by role.

---

3. Employee Management (HR Only)

🔹 Endpoints

3.1 Assign Employee to Manager

```
PUT /api/employees/{employeeId}/assign-manager/{managerId}
```

Response:

Json

```
{
  "message": "Employee assigned to manager successfully."
}

```

Constraints:

✅ Employee can have **only one** manager.

✅ Manager role must be **validated** before assignment.

3.2 Get Employee Details

```
GET /api/employees/{employeeId}
```

Response:

Json

```
{
  "id": 1,
  "name": "John Doe",
  "manager": {
    "id": 2,
    "name": "Jane Manager"
```

```
  },
  "reviews": [
    {
      "reviewer": "Jane Manager",
      "rating": 4.5,
      "comments": "Great work!",
      "date": "2024-02-05"
    }
  ]
}
```

Constraints:

☑️ Only **HR and the assigned manager** can view employee details.

---

4. Performance Reviews (Manager & HR)

🔹 Endpoints

4.1 Submit Employee Review

`POST /api/reviews`

Request:

```json
{
  "employeeId": 1,
  "reviewerId": 2,
  "rating": 4.5,
  "comments": "Excellent teamwork and problem-solving skills."
}
```

Response:

```json
{
  "message": "Review submitted successfully."
}
```

Constraints:

☑️ Only **HR or the assigned manager** can review an employee.

☑️ **Rating must be between 1 and 5**.

4.2 Get Employee Reviews

`GET /api/reviews/{employeeId}`

Response:

```json
{
  "reviews": [
    {
      "reviewer": "Jane Manager",
      "rating": 4.5,
      "comments": "Excellent teamwork!",
      "date": "2024-02-05"
    }
  ]
}
```

Constraints:

✅ Employee can view **only their own** reviews.

---

5. Self-Appraisal (Employee Only)

🔹 Endpoints

5.1 Submit Self-Appraisal

`POST /api/self-appraisal`

Request:

```json
{
  "rating": 4.0,
  "comments": "I contributed to multiple projects and improved efficiency."
}
```

Response:

```json
{
  "message": "Self-appraisal submitted successfully."
}
```

Constraints:

✅ Employees can **submit only one** appraisal per cycle.

---

6. Reporting & Analytics

🔹 Endpoints

6.1 Get Performance Reports (HR Only)

```
GET /api/reports?sort=top-performers&limit=5
```

Response:

```
{
  "topPerformers": [
    {
      "employee": "John Doe",
      "averageRating": 4.7
    }
  ]
}
```

Constraints:

✅ Only **HR** can access reports.

Note

✅ **Role-Based Authorization** (HR, Manager, Employee).

✅ **Input Validation**

✅ **Error Handling**

✅ **Database Transactions** to maintain consistency.

Final Deliverables

- ✅ **Fully documented API** (Swagger or Postman Collection).
- ✅ **Structured error handling**.
- ✅ **Database migration scripts**.