



**BHASKARACHARYA NATIONAL INSTITUTE FOR SPACE
APPLICATIONS AND GEO-INFORMATICS**
WEEKLY PROGRESS REPORT (29/12/2025 – 04/12/2025)

WEEK 4

PROJECT NAME

DIGITAL TOWN PLATFORM

PROJECT DESCRIPTION :

**DESIGN TO PROVIDE TOWN-LEVEL
INFORMATION, SERVICES, THROUGH A SINGLE
SYSTEM**

GROUP MEMBER :

Vandit Muniya, Mitanshu Parmar, Dixit Ram

GROUP ID :

2025G152

GROUP GUIDE :

Dr. MANOJ PANDYA

GROUP COORDINATOR :

SIDHDHARTH PATEL

COLLEGE GUIDE NAME :

NEHA FOTEDAR

COLLEGE GUIDE No. :

NOT ASSIGNED

COLLEGE GUIDE EMAIL. :

Nehafotedar.ce@ddu.ac.in

COLLEGE NAME :

DHARMSINH DESAI UNIVERSITY, NADIAD

| | |
|------------|--|
| 29/12/2025 | <ul style="list-style-type: none"> Vandit Muniya: Participated in detailed group discussion to finalize the database design and implemented the initial set of 9 MongoDB models, covering core entities such as user, services, marketplace items, feedback, news, and notifications. Mitanshu Parmar: Assisted in reviewing model schemas and validating relationships between entities to ensure alignment with project requirements and future scalability. Dixit Ram: Contributed to designing the ER diagram based on finalized schemas, ensuring correct entity relationships and normalization. |
| 30/12/2025 | <ul style="list-style-type: none"> Vandit Muniya: Extended the database by adding 3 additional models (marketplace category, service category, and transaction) required for proper categorization and transaction tracking. Mitanshu Parmar: Updated and refined the ER diagram to incorporate newly added models and verified consistency with existing schemas. Dixit Ram: Reviewed all models and ER relationships to ensure they follow standard database design practices. |
| 31/12/2025 | <ul style="list-style-type: none"> Vandit Muniya: Started the backend coding phase, setting up the project structure and implementing core logic for authentication and authorization. Mitanshu Parmar: Developed authentication middleware to handle user verification and secure access to protected routes. Dixit Ram: Implemented RBAC (Role Based Access Control) middleware to enforce role specific permissions across the system. |
| 01/12/2025 | <ul style="list-style-type: none"> Vandit Muniya: Created essential API routes, including authentication routes and test routes, to support user login and role-based operations. Mitanshu Parmar: Developed the authentication controller to handle login, role switching, and token-based access. Dixit Ram: Integrated middleware with routes and verified end-to-end request flow between controllers and database. |

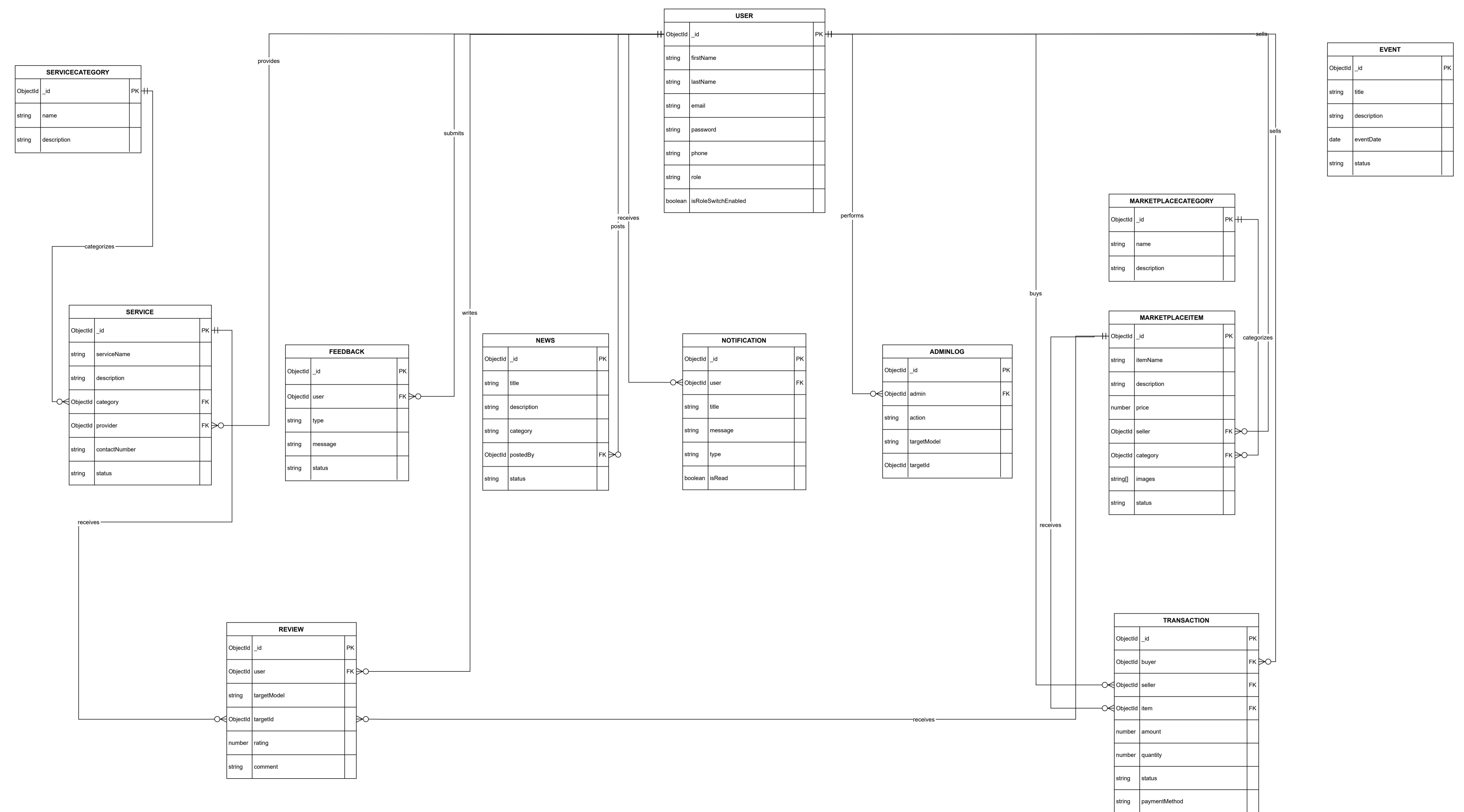
| | |
|------------|---|
| | |
| 02/12/2025 | <ul style="list-style-type: none"> Vandit Muniya: Performed API testing using Postman, validating request response flow for authentication and protected routes. Mitanshu Parmar: Identified and fixed logical and authorization related bugs found during testing. Dixit Ram: Assisted in debugging middleware and routes, ensuring stable execution and correct role enforcement. |
| 03/12/2025 | All absent(Reporting at college) |
| 04/12/2025 | Holiday (Sunday) |

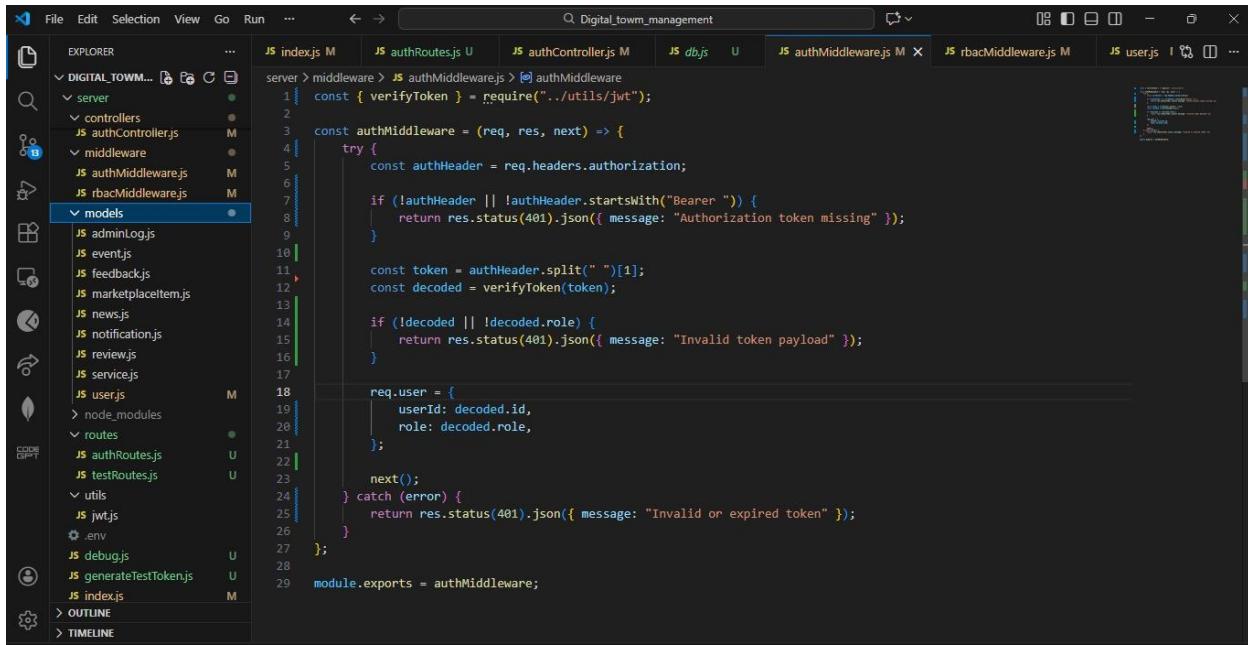
| | |
|-----------------------|---|
| Next Week Plan | <ul style="list-style-type: none"> Frontend development : Creation of Login and Signup pages with validation and authentication flow, development of Home page as a central dashboard Separate pages for all major routes such as Services, Events, Marketplace, Feedback, Profile, etc. Role-based UI rendering using React Backend development : Creation of routes for all modules (services, events, news, marketplace, feedback, reviews, etc.) Integration of routes with existing authentication and RBAC middleware |
|-----------------------|---|

Github link:

https://github.com/MeetanshuP/Digital_Town_Management

SCREEN SHOTS :





```
server > middleware > JS authMiddleware.js > authMiddleware
1 | const { verifyToken } = require("../utils/jwt");
2 |
3 | const authMiddleware = (req, res, next) => {
4 |   try {
5 |     const authHeader = req.headers.authorization;
6 |
7 |     if (!authHeader || !authHeader.startsWith("Bearer ")) {
8 |       return res.status(401).json({ message: "Authorization token missing" });
9 |     }
10 |
11 |     const token = authHeader.split(" ")[1];
12 |     const decoded = verifyToken(token);
13 |
14 |     if (!decoded || !decoded.role) {
15 |       return res.status(401).json({ message: "Invalid token payload" });
16 |
17 |     req.user = {
18 |       userId: decoded.id,
19 |       role: decoded.role,
20 |     };
21 |
22 |     next();
23 |   } catch (error) {
24 |     return res.status(401).json({ message: "Invalid or expired token" });
25 |   }
26 |
27 | };
28 |
29 | module.exports = authMiddleware;
```