

Documentation of the Project UpToDate Assignment 02

1. Project Overview:

In this milestone, we have developed a strong backend for UpToDate project which is built using Node.js, Express.js, and MongoDB. The backend features provide:

- User Authentication (Register/Login).
- User Data Management (Bookmarks, Preferences).
- News API Integration (Fetching news articles).
- CRUD Operations for managing user-related data.

2. Folder Structure & Purpose

a. Root Directory (/)

server.js → Main server file, initializes Express, connects to MongoDB, and defines routes.

app.js → Entry point for the application.

.env → Stores environment variables like database connection string.

package.json → Lists project dependencies and scripts.

package-lock.json → Dependency lock file.

README.md → Documentation for project setup and usage.

request.rest → API testing file with sample requests.

b. config/ (Configuration Files)

db.js → MongoDB connection logic.

c. models/ (Database Schemas)

User.js → Defines the user schema, including username, email, and hashed password.

UserData.js → Stores user preferences and bookmarks.

d. routes/ (API Endpoints & Business Logic)

index.js → Default route file for the homepage.

newsRoutes.js → Placeholder for handling news fetching.

userDataRoutes.js → Manages CRUD operations for user data (bookmarks & preferences).

userRoutes.js → Handles user authentication (Register & Login).

users.js → Example user listing route.

e. views (Frontend Templates - EJS & Jade)

index.ejs → Main template file for rendering pages.

index.jade → Alternative template file for layout.

layout.jade → Base layout template.

error.jade → Error handling page.

f. Other Folders

bin/ → (typically for startup scripts.)

docs/ → Documentation files.

frontend/ → Placeholder for frontend code.

3. API Endpoints

a. User Authentication (/api/users)

Method	Endpoint	Description
POST	/api/users/register	Register a new user.
POST	/api/users/login	User login.

b. User Data Management (/api/user-data)

Method	Endpoint	Description
POST	/api/user-data/add	Adds bookmarks and preferences for a user.
GET	/api/user-data/UserId/:userId	Retrieves stored user data.

PUT	/api/user-data/update/:userId	Updates bookmarks and preferences.
DELETE	/api/user-data/delete/:userId	Deletes user data from the system.

c. News API (/api/news)

Method	Endpoint	Description
GET	/api/news/	Fetches news articles (currently a placeholder).

4. Technologies Used

- **Backend** → Node.js, Express.js
- **Database** → MongoDB (Mongoose ORM)
- **Authentication** → Bcrypt for password hashing
- **API Testing** → REST Client (request.rest)

5. Team Contributions

The work was evenly distributed where main focus for each was on:

Name	Contributions
Harshiv Bharat Patel	Implemented CRUD Operations for User Data
Saugat Pokharel	Developed User Authentication (Register, Login)
Abhishek Das	Designed schema for the database and worked on Documentation

6. Future Improvements

- Implement JWT Authentication instead of sessions.
- Complete News API functionality.
- Enhance Error Handling & Logging.
- Add Unit Tests for better reliability.