# **Capstone Project Documentation – VAS Feedback & Rating System**

## 1. Project Overview

****Title & Tag-line:****

****VAS Feedback & Rating System**** – A web-based platform to collect user feedback and ratings for Value-Added Services (VAS) offered by a telecom operator.

****Objective:****  
The system provides customers with an easy-to-use platform to rate and review VAS services, helping the operator track customer satisfaction and identify areas of improvement.

****Tech Stack:****

****Frontend:**** React, HTML, CSS, JavaScript, Axios

****Back-end:**** Node.js (or API endpoint integration if provided)

****Database:**** MySQL

****Tools:**** Git, npm, Docker

## Architecture

The system follows a ****full-stack architecture**** with ****React frontend**** integrated with a ****Node.js/Express backend**** through REST APIs. The backend is responsible for processing data, storing feedback, and providing analytics endpoints. APIs were tested using ****Postman**** to ensure reliability before frontend integration.

### Components:

****React Frontend**** – Handles user interactions, feedback submission forms, star rating UI, and displays feedback reports.

****Back-end (Node.js/Express)**** – Provides RESTful APIs for feedback submission, retrieval, and analytics.

****Feedback API**** – Receives data from the frontend, processes it, and communicates with the database.

****Rating Analytics Module**** – Calculates average ratings, generates service-wise reports, and provides data for dashboard visualizations.

****Database (MySQL)**** – Stores user feedback, service details, and rating analytics.

### High-Level Workflow Diagram:

User → React Frontend → Backend (Node.js/Express) → Database → Analytics Module → React Frontend (Reports)

### Testing:

****Postman API Testing****: All endpoints were tested in Postman for correctness and error handling before integrating with the frontend.

****Frontend Integration****: React frontend consumes these APIs using Axios to ensure a seamless user experience.

## Feedback Service Documentation

## 4. Frontend (React)

****Pages & Components:****

1. Register page
2. OTP Verification page
3. Login Page
4. User dashboard page
5. Service page
6. Reviews page
7. Feedback form
8. Admin portal page

****State Management:****

React Hooks: useState, useEffect

****API Calls:****  
Axios is used for REST API integration:

axios.get('/api/feedback')

.then(response => setFeedbackList(response.data))

.catch(error => console.error(error));

****Rating System Example:****  
Users can select a star rating (1–5) and submit comments via a form.

## 5. Deployment Setup

### Local Run Instructions

Clone repository: git clone <repo\_url>

Install dependencies: npm install

Start frontend: npm start

### Backend Setup (if separate):

Start API server: node server.js or npm run dev

### Docker Setup (Future Enhancement):

docker build -t vas-frontend .

docker run -p 3000:3000 vas-frontend

### Jenkins CI/CD (Future Enhancement):

Pipeline: ****Build → Test → Deploy****

## Cloud/DevOps

****Hosting:**** Future deployment on AWS S3/EC2 or Docker containers

****Scaling:**** Load balancer integration for backend services

## Testing

****Unit Tests:**** React Testing Library for components

****API Tests:**** Postman collections for feedback endpoints

****UI Tests:**** Manual validation of forms, ratings, and review page.

## 8. Challenges & Learnings

****CORS Issues:**** Resolved by enabling CORS in backend API

****API Integration:**** Ensuring correct endpoint URLs with environment variables

****State Management:**** Handling asynchronous data in React with Axios

## 9. Future Enhancements

Include ****data visualization**** for analytics dashboards

Enable ****email/SMS notifications**** for submitted feedback

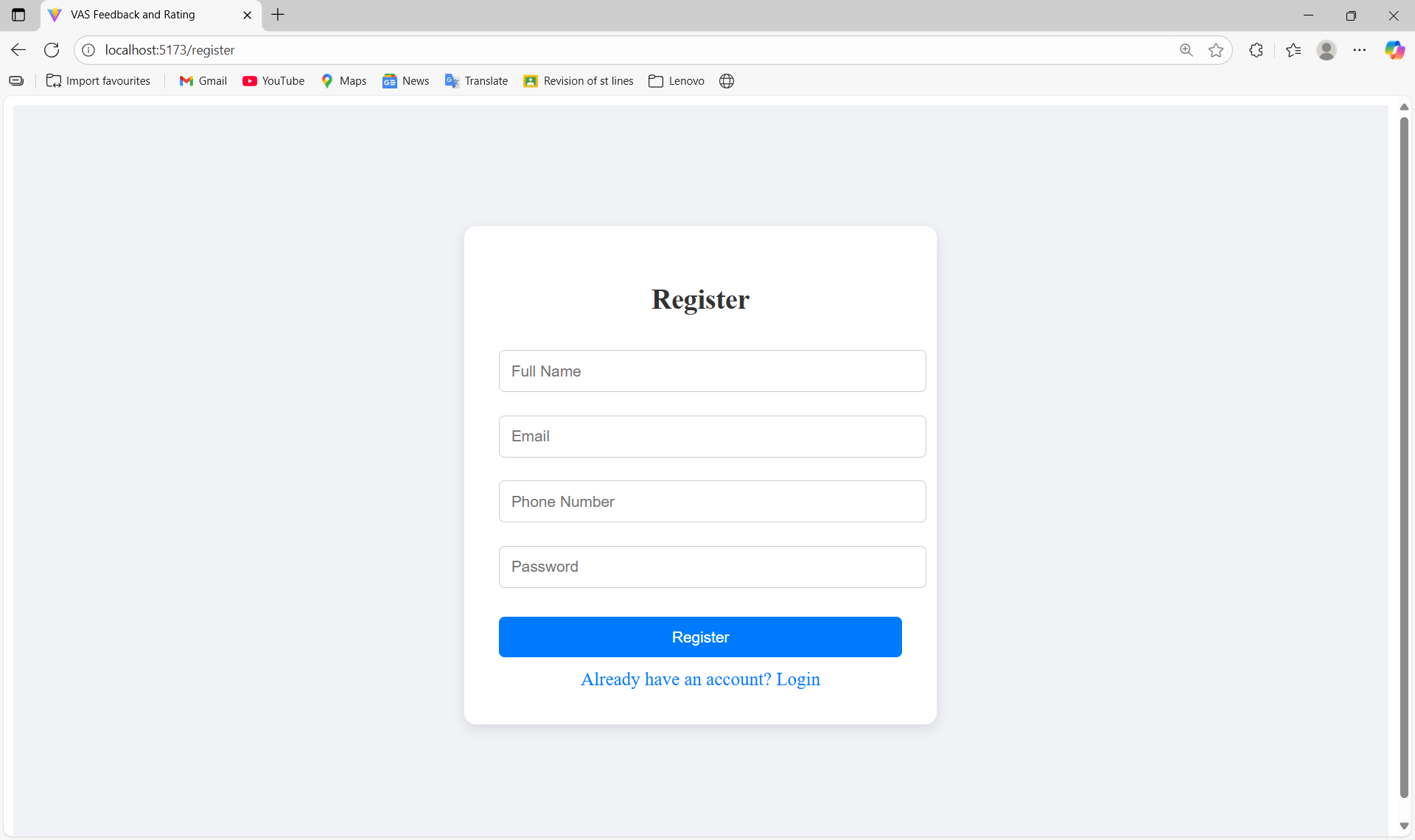
**Pagination & Lazy Loading f**or large datasets,add **pagination or infinite scrolling** to improve UI performance and user experience.

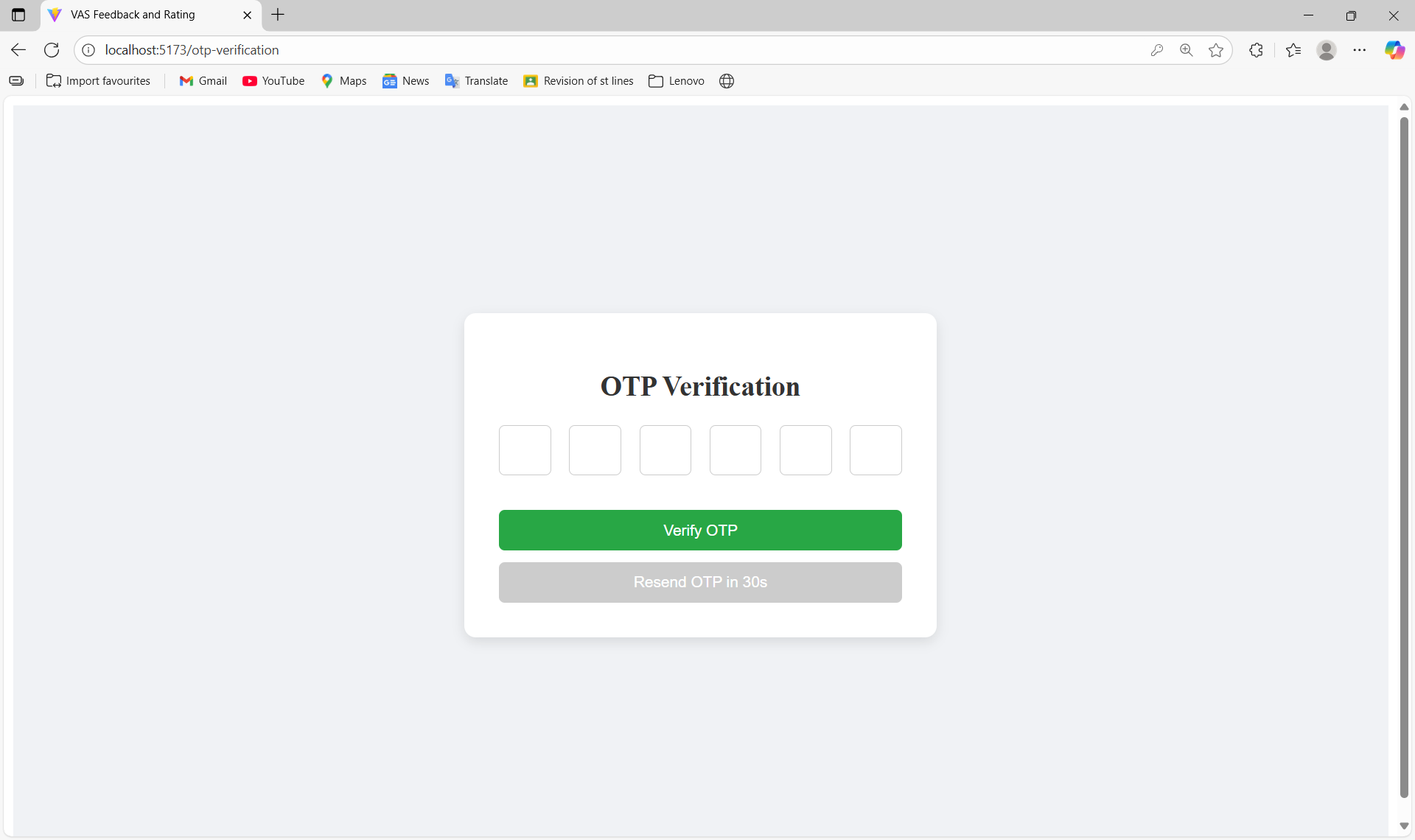
Use **AI/ML libraries** to analyze user comments and generate sentiment scores (positive, negative, neutral).

## 10. Appendix

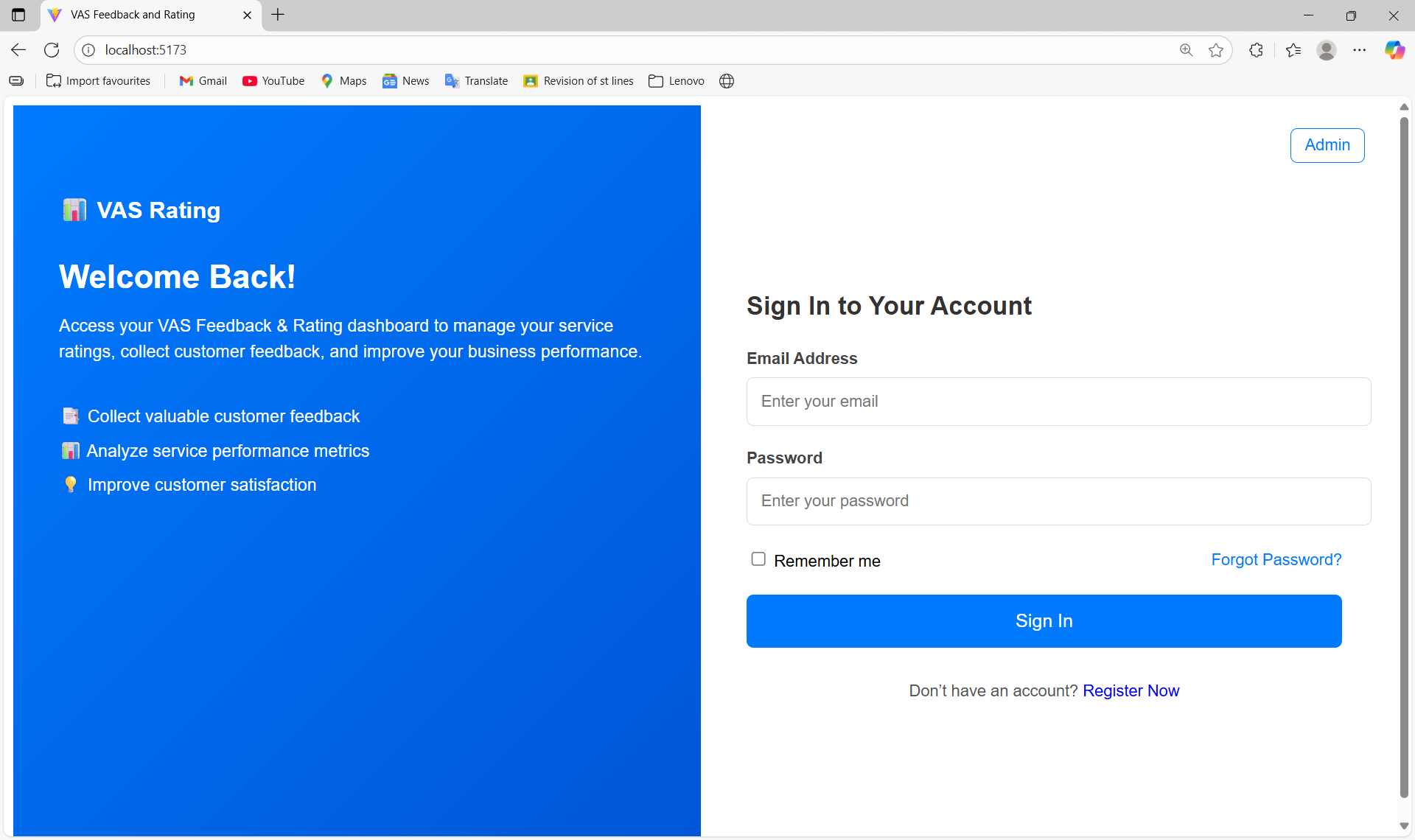
****GitHub Repository:**** [febiiinn (Febin Phillips )](https://github.com/febiiinn)

****UI Screenshots:**Register Page** - New User Registration

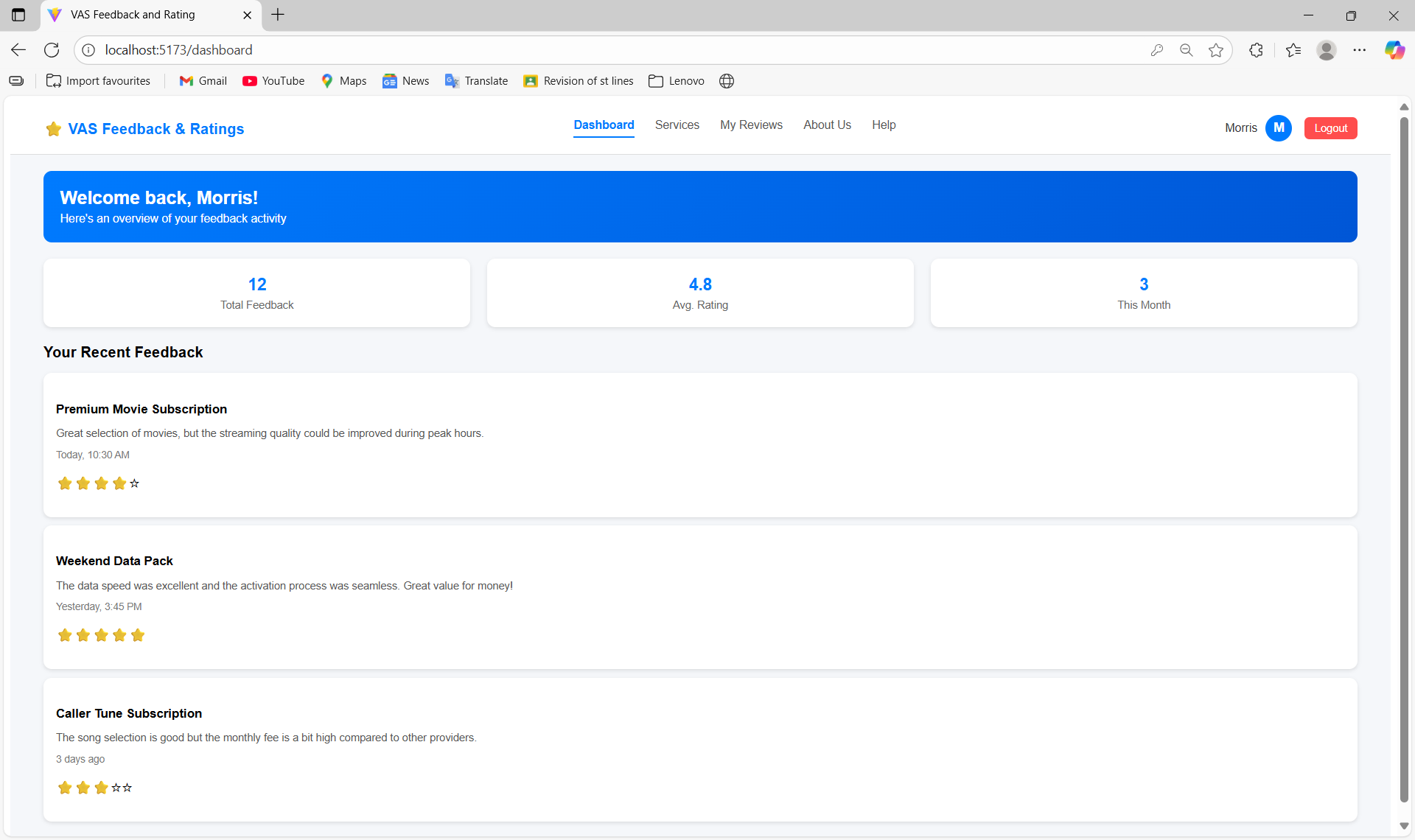




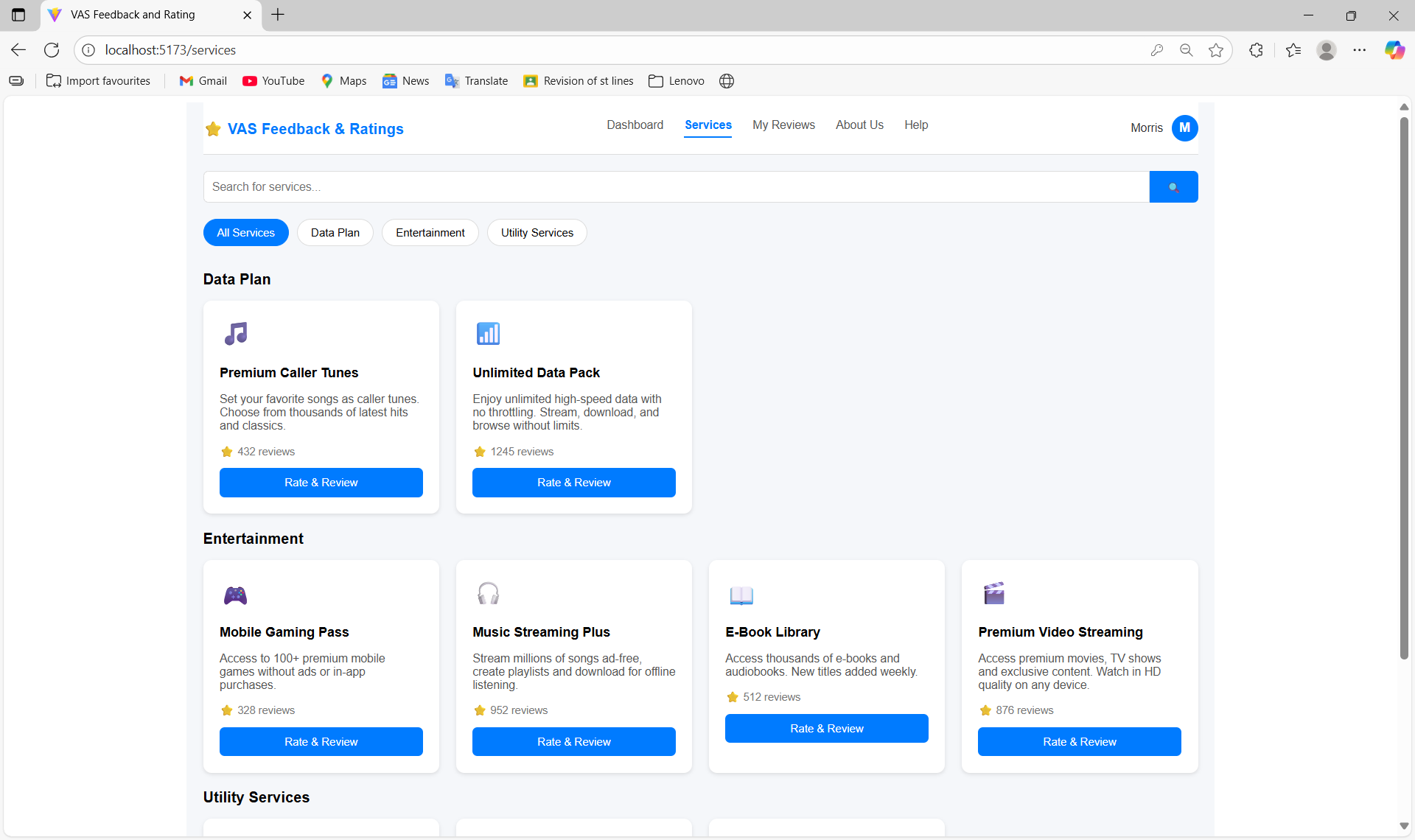
****Login Page**** – User login



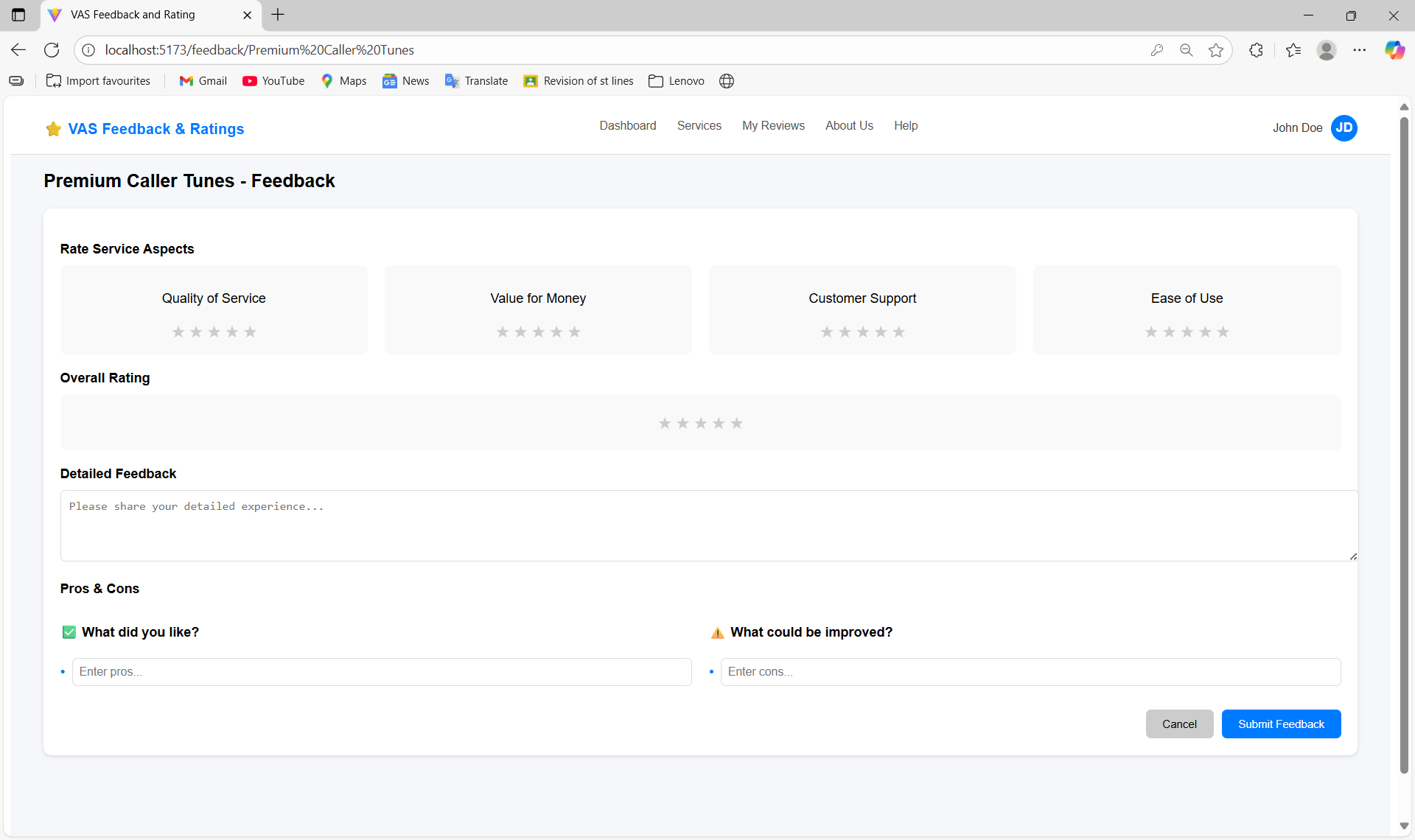
**UserDashboard Page** -



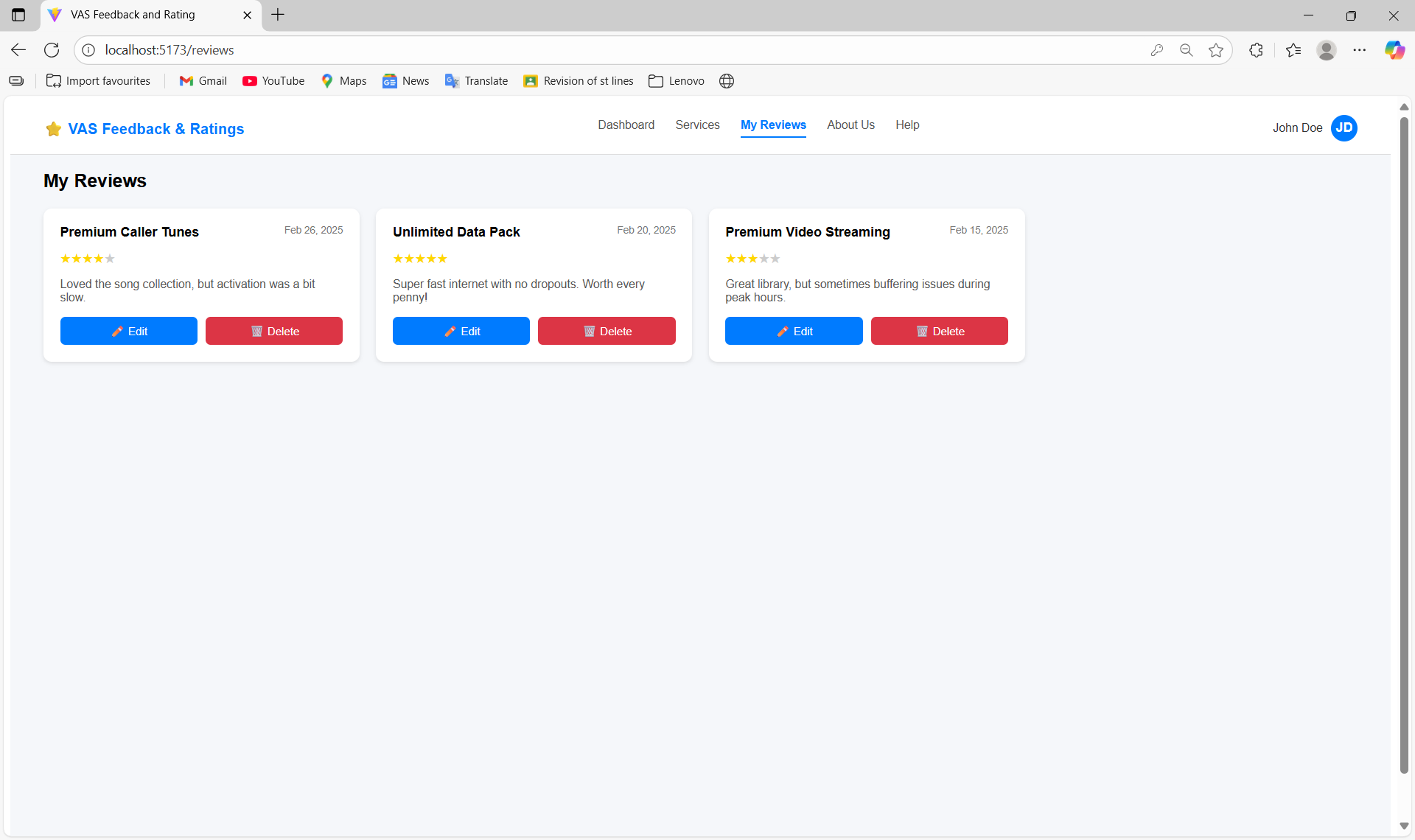
****Services Page**** – List of available VAS services



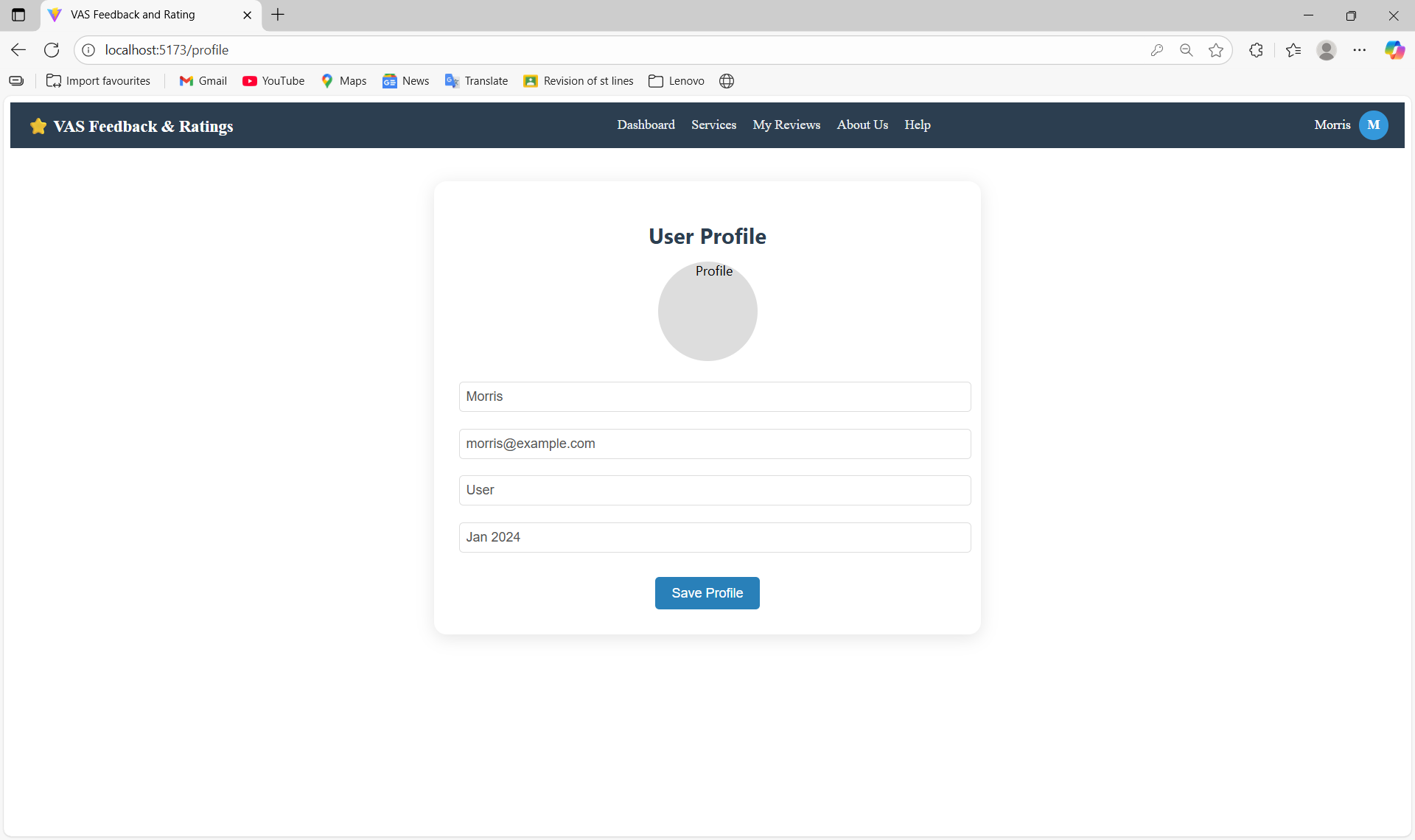
****Feedback Page**** – Submit ratings and comments



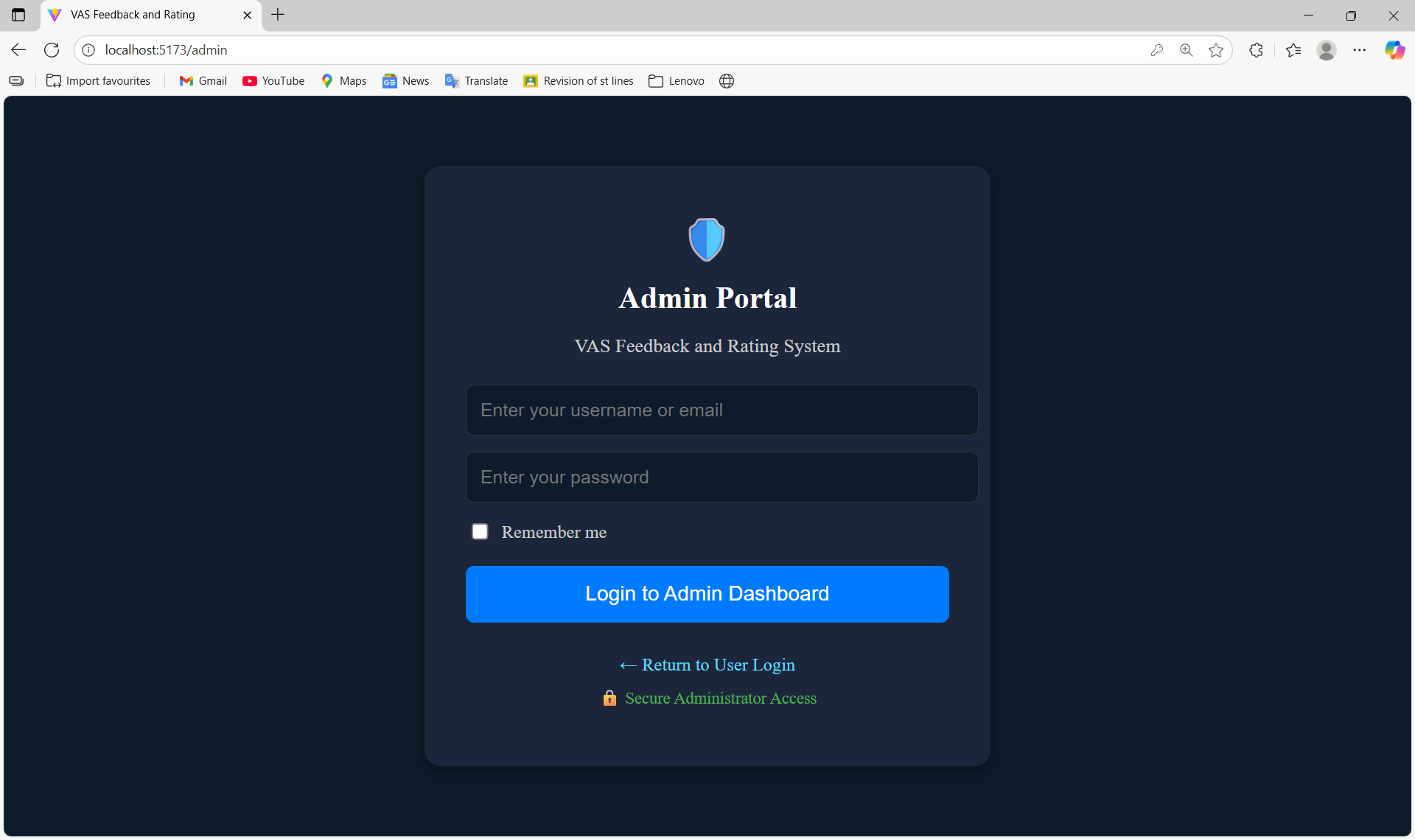
****Reviews Page**** – View all feedback and ratings



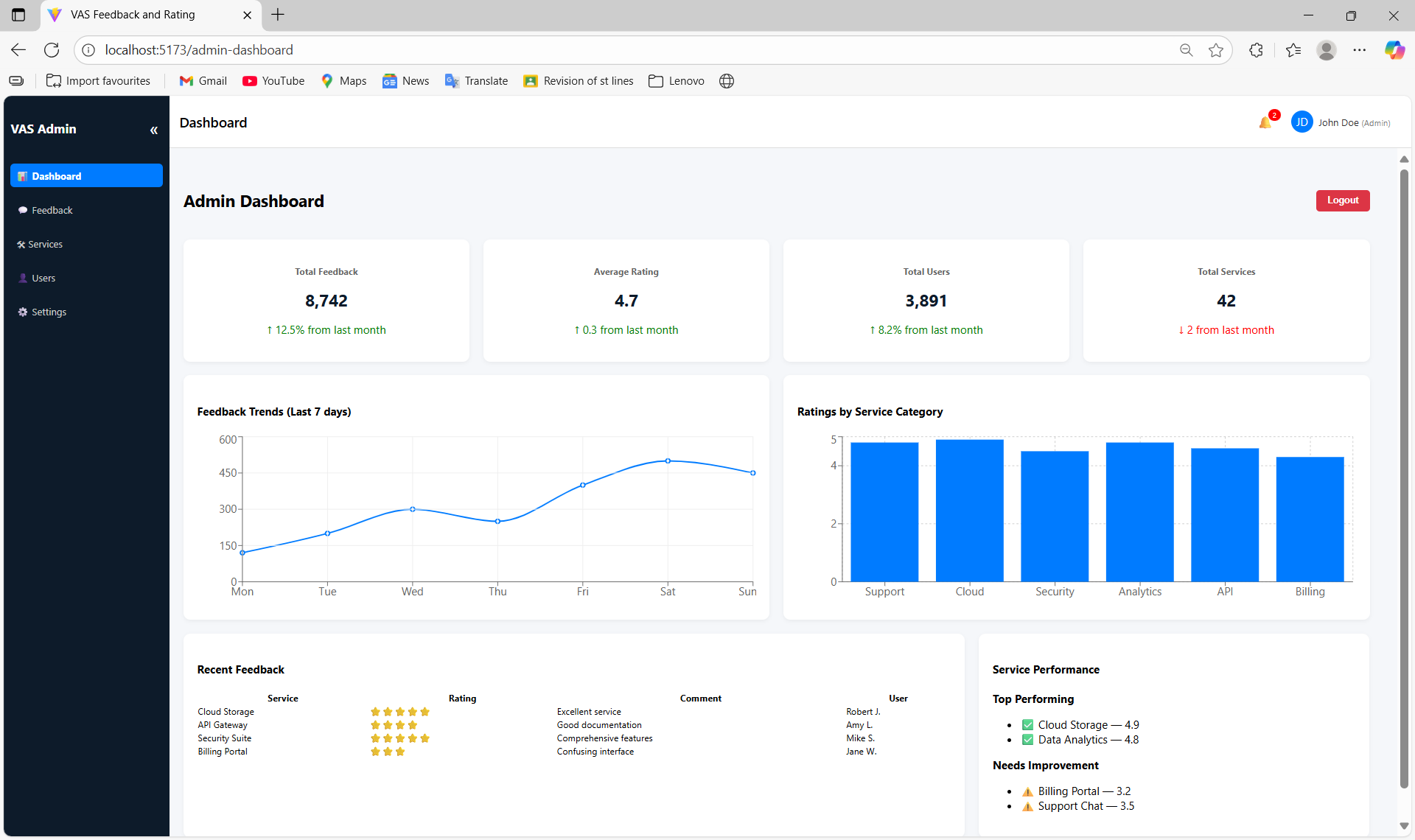
**User Profile Page -** Users can edit their profile

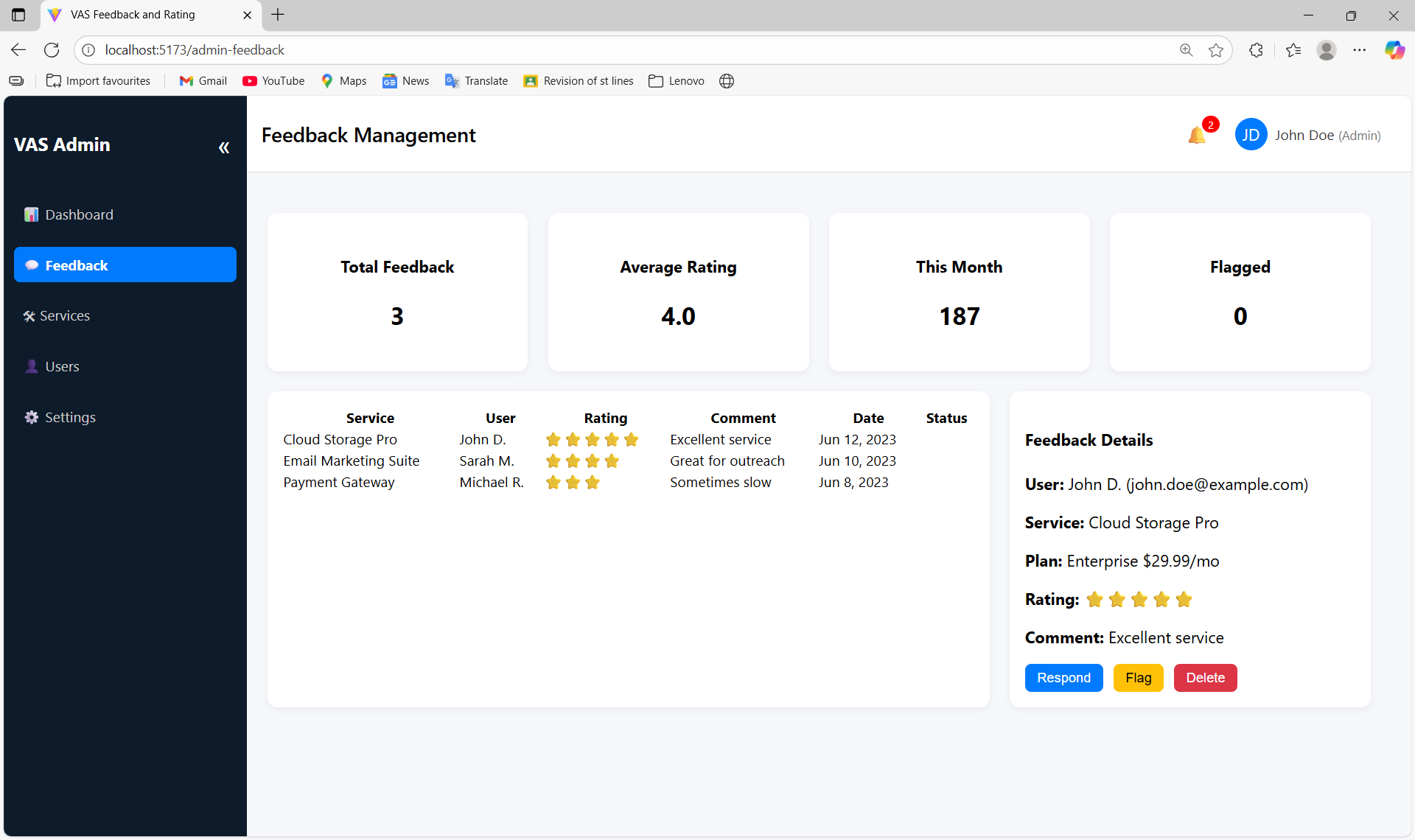


**Admin Login Page-**

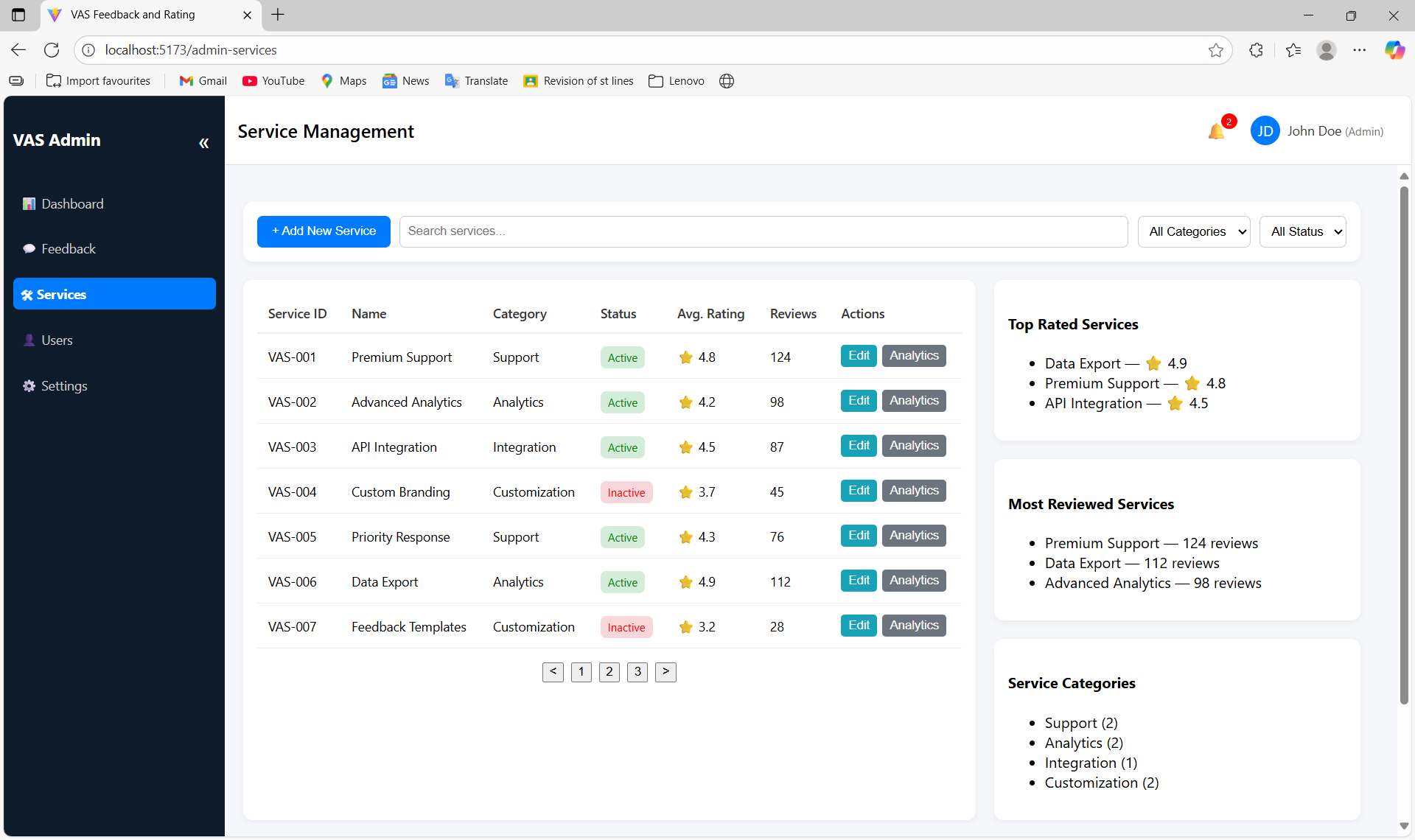


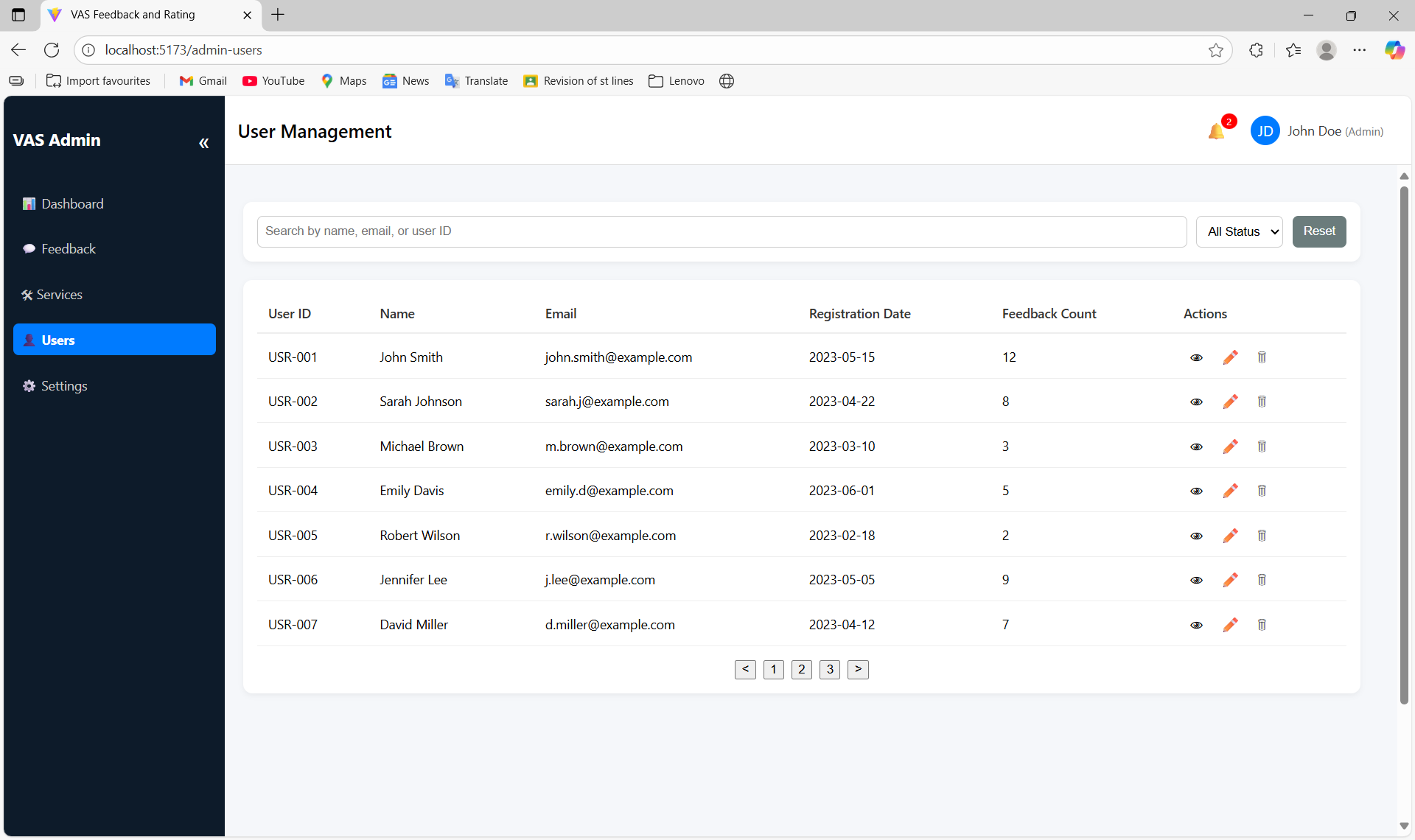
**Admin Dashboard Page-**

  
  
**Feedback Management Page-**

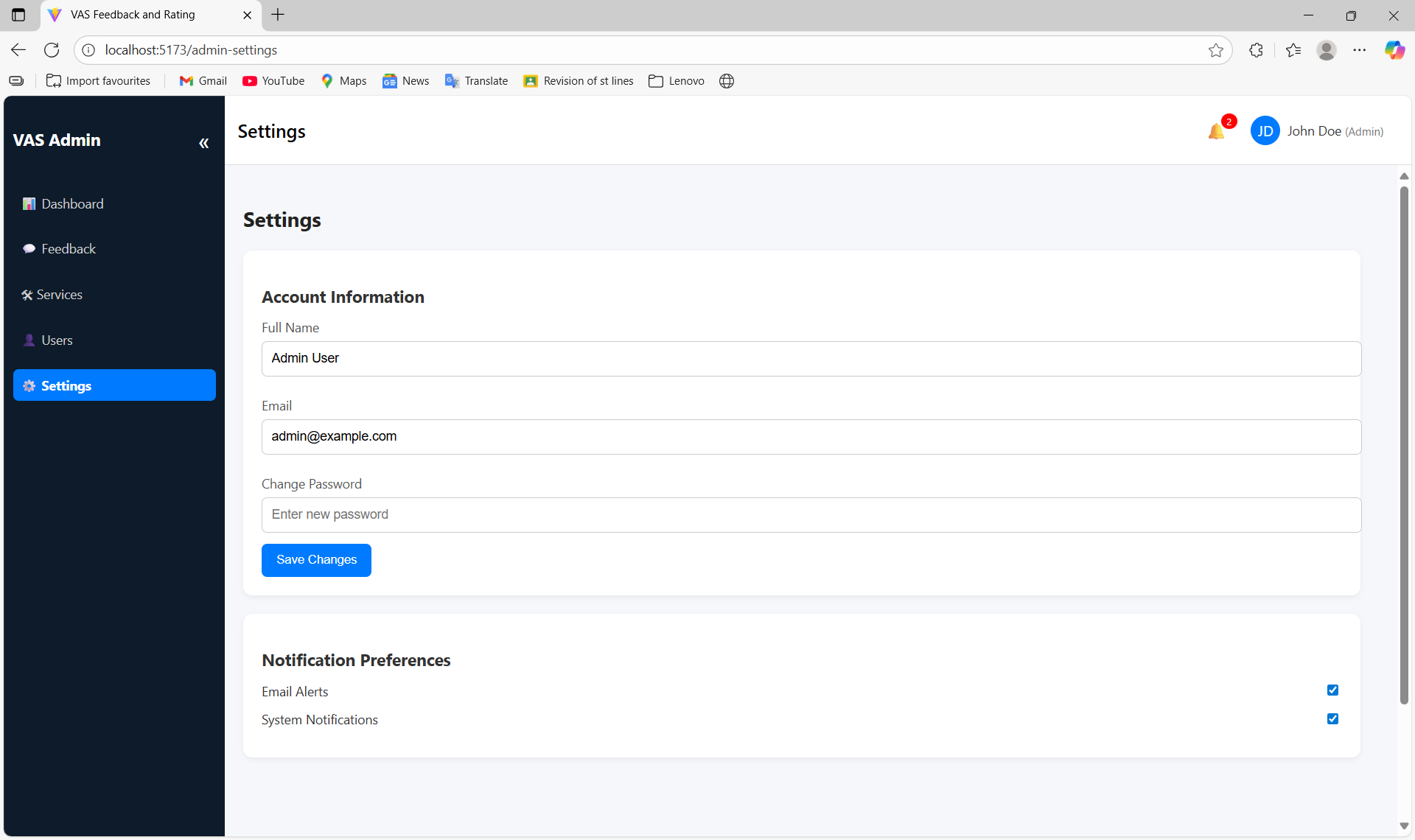


**Service Management Page-**

  
  
**User Management Page-**



**Admin Settings Page-**



****References:**** React docs, Axios docs, Node.js docs