Project Report

**Project Report: Unicode – Unified Developer Profile Platform**

**🧩 Problem Statement**

In the modern digital landscape, developers often maintain multiple profiles across various coding platforms like GitHub, LeetCode, and CodeChef. This fragmentation poses challenges:

* **Scattered Profiles**: Difficulty in presenting a cohesive view of one's coding journey.
* **Inefficient Sharing**: Sharing multiple links is cumbersome for networking or job applications.
* **Manual Updates**: Keeping each profile updated individually is time-consuming.

**Unicode** addresses these issues by providing a unified platform that aggregates and showcases a developer's profiles in a consolidated manner.

**🎯 Project Objective**

To develop a full-stack web application that:

* **Aggregates Developer Profiles**: Integrates data from platforms like GitHub, LeetCode, and CodeChef.
* **Provides a Unified View**: Offers a single, shareable link representing the developer's coding journey.
* **Ensures Security and Scalability**: Implements secure authentication and is scalable for future enhancements.
* **Facilitates Continuous Integration/Deployment**: Streamlines development and deployment processes.

**🛠️ Technology Stack**

**Frontend**

* **React.js**: For building dynamic user interfaces.
* **Tailwind CSS**: For efficient and responsive styling.
* **React Router**: To manage client-side routing.
* **React Icons**: For incorporating scalable vector icons.

**Backend**

* **Python & Django**: For robust backend development.
* **Django REST Framework (DRF)**: To create RESTful APIs.
* **JWT Authentication**: For secure user authentication.
* **PostgreSQL**: As the production database.

**Deployment & Hosting**

* **Frontend**: Deployed on [Vercel](https://vercel.com/), facilitating seamless CI/CD.
* **Backend**: Hosted on **AWS EC2**, providing scalability and control.
* **Domain Management**: Utilized [No-IP](https://www.noip.com/) for free Dynamic DNS services.
* **SSL Certification**: Implemented HTTPS using Let's Encrypt for secure data transmission.

**🔄 System Architecture Overview**

plaintext

CopyEdit

[User Browser]

|

v

[Frontend (Vercel) - React.js]

|

v

[API Calls]

|

v

[Backend (AWS EC2) - Django REST API]

|

v

[Database - PostgreSQL]

**🧪 Feature Flow Description**

1. **User Registration/Login**: Secure authentication using JWT.
2. **Profile Integration**: Users can link their GitHub, LeetCode, and CodeChef profiles.
3. **Data Aggregation**: The system fetches and consolidates data from linked profiles.
4. **Unified Profile Display**: Presents a comprehensive view of the user's coding journey.
5. **Shareable Link**: Users receive a unique link to share their unified profile.

**☁️ Deployment Details**

**Frontend on Vercel**

* **CI/CD Integration**: Automatic deployments triggered by GitHub commits.
* **Custom Domain Setup**: Configured for personalized URLs.
* **Performance Optimization**: Leveraged Vercel's global CDN for faster load times.

**Backend on AWS EC2**

* **Server Configuration**: Set up using Ubuntu, with Nginx and Gunicorn for serving the Django application.
* **Database Management**: PostgreSQL configured for data persistence.
* **Security Measures**: Implemented firewalls and security groups to protect the server.

**Domain and SSL**

* **Dynamic DNS**: Used No-IP to manage dynamic IP addresses and provide a consistent domain name.
* **SSL Certification**: Employed Let's Encrypt to secure the application with HTTPS.

**📚 Learning Outcomes**

**1. AWS EC2 Deployment**

* Gained hands-on experience in setting up and configuring virtual servers.
* Learned to manage server security, networking, and application deployment.

**2. Continuous Integration/Continuous Deployment (CI/CD)**

* Understood the importance of automating the deployment process.
* Implemented CI/CD pipelines to streamline development workflows.

**3. Django Backend Development**

* Developed RESTful APIs using Django and DRF.
* Implemented JWT for secure user authentication.

**4. Domain Management with DDNS**

* Learned to configure and manage domain names using No-IP.
* Understood the workings of Dynamic DNS and its applications.

**5. SSL Certificate Implementation**

* Acquired knowledge on securing web applications using SSL certificates.
* Implemented HTTPS to ensure secure data transmission.

**🔗 Live Application Links**

* **Frontend**: <https://unicode-ochre.vercel.app/>
* **Backend Admin Panel**: <https://unicodebackend.ddns.net>