National AI Institute Website
Technical Design Document

1. Tech Stack

1.1 Development Stack

- **Backend Language**: Python
- **Web Framework**: FastAPI
- **Database**: MySQL Server
- **Version Control**: Git
- **Libraries/Tools**:
  - Python 3.x
  - FastAPI 0.68.x
  - MySQL 8.x
  - Git 2.x
- **IDE**: Any code editor or IDE preferred by the development team.
- **API Testing**: Postman for API testing.
- **System Requirements**: The development machines should have at least 8GB of RAM and sufficient disk space for development purposes. Specific processor power requirements are not critical for development.

1.2 Deployment Stack

- **Production Environment**: AWS (Amazon Web Services)
- **Server Configuration**: EC2 instances with appropriate specifications for production usage.
- **Database**: MySQL Server (AWS RDS)
- **Version Control**: Git (GitHub or AWS CodeCommit)
- **Libraries/Tools**:
  - Python 3.x
  - FastAPI 0.68.x
  - MySQL 8.x
  - Git 2.x
- **API Testing**: Postman for API testing.
- **System Requirements**: The production servers should meet the performance requirements specified by AWS for hosting FastAPI applications and MySQL databases.

2. Accounts and Infrastructure

2.1 Development

- **Development Environment**: Local development machines for each developer.
- **Version Control Repository**: GitHub.
- **AWS Development Account**:
  - **URL**: AWS Console
  - **Account Name**: [Our AWS Account Name]
  - **Login Credentials**: [Developer's AWS IAM credentials]
- **Postman Account:**
  - **URL**: Postman
  - **Account Name**: [Our Postman Account Name]
  - **Login Credentials**: [Developer's Postman credentials]

2.2 Production

- **Production Environment**: AWS EC2 instances and RDS for MySQL.
- **AWS Production Account**:
  - **URL**: AWS Console
  - **Account Name**: [Our AWS Account Name]
  - **Login Credentials**: [Production Team's AWS IAM credentials]

Data Sources, Models, Timing

1.1 Data Sources

- Data sources for the National AI Institute website include:
  - User-generated content (user profiles, comments, etc.)
  - Administrative data (website settings, configurations)
  - External data feeds (e.g., research data, news)
- Data will be created and maintained by website users and administrators.

1.2 Data Models and Structure

- **Database Tables**:
  - Users
  - Resources
  - Information
  - News

    - Projects
    - Admin
    - Research Data
    - …

- Detailed information about each table and its columns will be documented separately and will be revised if necessary.
- Data interchange format: JSON for API responses.
- Working ER Diagram: https://drive.google.com/file/d/16VotRB7SHvZmnnG06Pp9zj8vKO9jq-uS/view?usp=sharing

1.3 API Contracts

https://docs.google.com/document/d/1s62DqrhjWhqDYCpItxfeKUq4McNPXJzYKuXc4rQhvfQ/edit

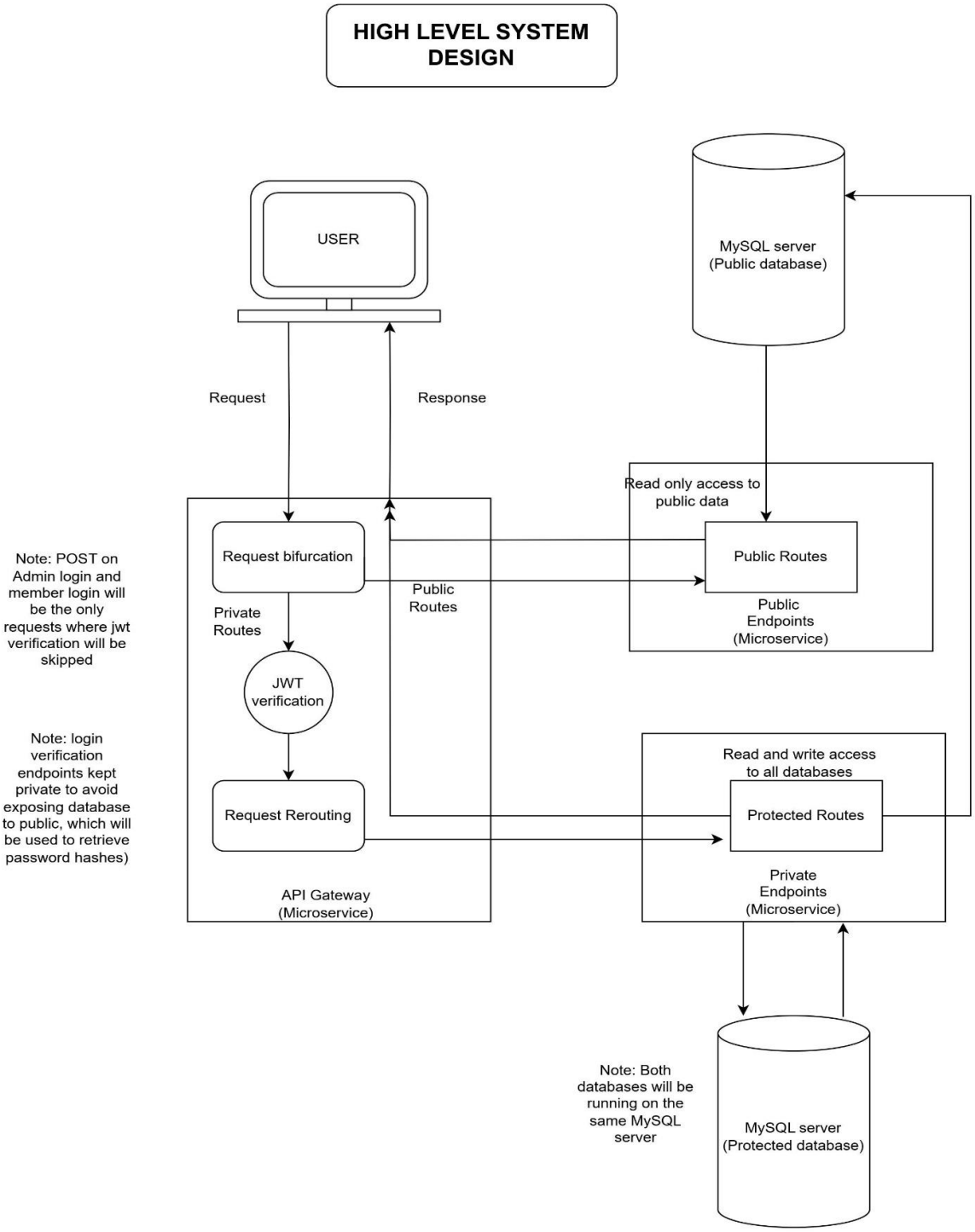
1.4 GitHub Repository Link

https://github.com/xlab-classes/cse611-fall-2023-team-AI4ExceptionalEd_Website_Private

1.5 Timing

- Imports and Exports: Data imports from external sources will be scheduled periodically (e.g., daily, or weekly).
- Data Retention: User-generated data will be retained indefinitely, while external data may vary in retention period.
- Data Removal: Deleted data will be archived for a specified duration before permanent deletion.

System Architecture Diagram



Deployment Methodology

The National AI Institute website will be deployed to the AWS production environment using the following methodology:

1. Code will be pushed to the main branch in the Git repository.
2. AWS CodePipeline will be used for continuous integration and deployment.
3. Application and database infrastructure will be provisioned using AWS CloudFormation or Terraform.
4. The database schema will be created or migrated using database migration tools.
5. Application code will be deployed to EC2 instances using containerization (e.g., Docker) or directly onto EC2 instances.
6. Monitoring and error tracking will be set up using AWS CloudWatch and third-party tools.
7. The website will be tested in the test environment and furthermore in the production environment.
8. If a turnover is required, data will be migrated from the development database to the production database.
9. The system will be monitored post-deployment for performance and issues.
10. Regular backups and disaster recovery procedures will be established.

Postman API Testing

API testing for the National AI Institute website will be conducted using Postman. Postman collections and requests will be used to test various API endpoints for functionality, security, and performance. The following steps outline the Postman testing process:

1. **Import Postman Collection**: The Postman collection containing API requests and tests will be imported into the Postman environment.
2. **Environment Configuration**: Development and production environment configurations will be set up in Postman to manage different API endpoints and variables.
3. **API Testing**: API requests will be executed in Postman to verify the functionality of different endpoints, including user registration, data retrieval, and administrative operations.
4. **Authorization and Authentication Testing**: Postman will be used to test authorization and authentication mechanisms, including token-based authentication for secure API access.
5. **Performance Testing**: Load testing and performance testing will be performed using Postman to assess how the API performs under various traffic conditions.
6. **Security Testing**: Postman will be used to conduct security testing, including testing for vulnerabilities such as SQL injection and cross-site scripting (XSS).
7. **Automated Testing**: Postman tests will be automated and integrated into the CI/CD pipeline to ensure that API functionality is continuously validated during development and deployment.

Postman test results and reports will be used to identify and address any issues or anomalies in the API behavior, ensuring the reliability and security of the National AI Institute website's API endpoints.