IT 334 – DevOps Engineering on AWS Cloud

Project – Building a Highly Available, Scalable Web Application

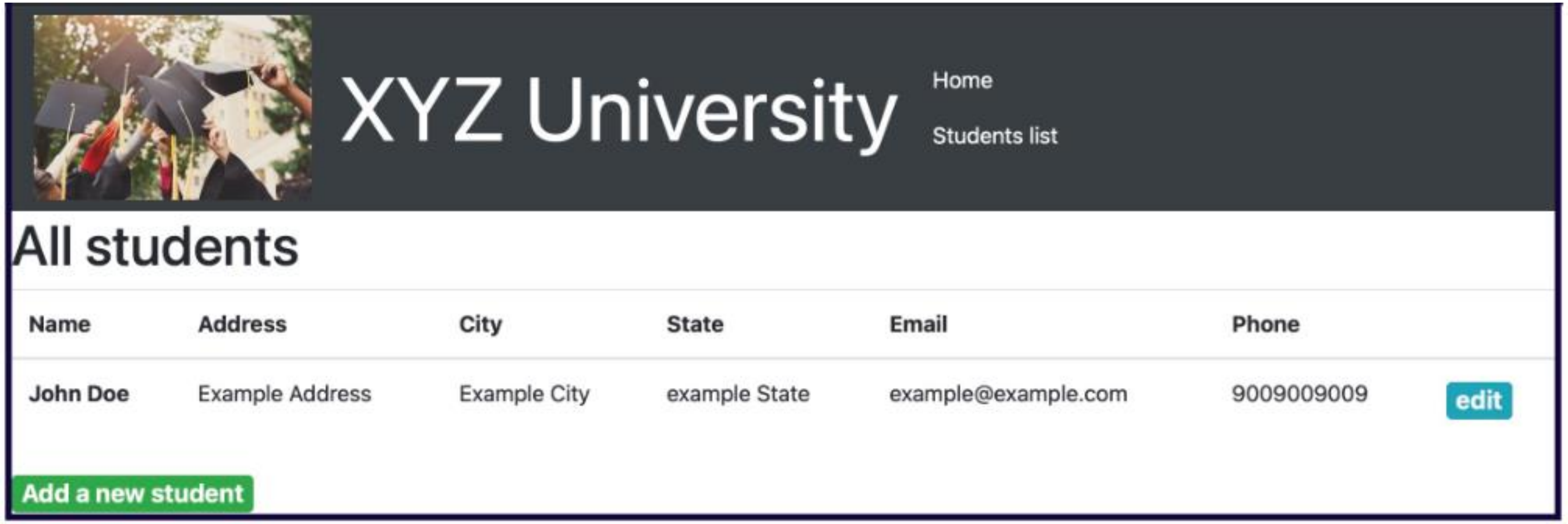
Group 13:
Lejla Muratović
Aldijana Čulezovvić

Date:
05/06/2024

Teacher:
Dzenana Dzevlan

Business scenario overview

- Example University is preparing for the new school year. The huge volume of inquiries has resulted in complaints to the admissions department over the slowness or unavailability of their web application for student records during the peak admissions period.
- The challenge is to plan, design, build, and deploy the web application to the AWS Cloud, which should support thousands of users, and be highly available, scalable, load balanced, secure, and high performing, during the peak admissions period
- The following image shows an example of the student records web application. Users can view, add, delete, and modify student records.



All students

Name	Address	City	State	Email	Phone	
John Doe	Example Address	Example City	example State	example@example.com	9009009009	edit

Business scenario overview

Solution Requirements:

- **Functional:** The solution should meet the functional requirements, such as the ability to view, add, delete, or modify the student records, without any perceivable delay.
- **Load balanced:** The solution should properly balance user traffic to avoid overloaded or underutilized resources.
- **Scalable:** The solution should be designed to scale to meet the demands that are placed on the application.
- **Highly available:** The solution should be designed to have limited downtime when a web server becomes unavailable.
- **Secure:**
    1. The database should be secured and can't be accessed directly from public networks.
    2. The web servers and database can be accessed only over the appropriate ports.
    3. The web application should be accessible over the internet.
    4. The database credentials shouldn't be hardcoded into the web application.
- **Cost optimized:** The solution should be designed to keep costs low.
- **High performing:** The routine operations should be performed without a perceivable delay under normal, variable, and peak loads.

Solution overview

High-level Description

- We created a secure and scalable web application on AWS to make sure it can effectively manage large amounts of traffic. With the ability to view, add, delete, and alter records, this program acts as a platform for Example University's student record management.

Design Considerations

1. **High Availability** → Multiple AZs
2. **Functionality** → Amazon EC2
3. **Scalability** → Amazon RDS and Auto Scalling
4. **Security** → VPC, Security Groups, Secrets Manager
5. **Load balancing** → Amazon ELB

Use Cases

- Administrative Access: Faculty members can access and modify student records
- Student Information Retrieval: Students can view their personal and academic information

Architecture diagram of the solution

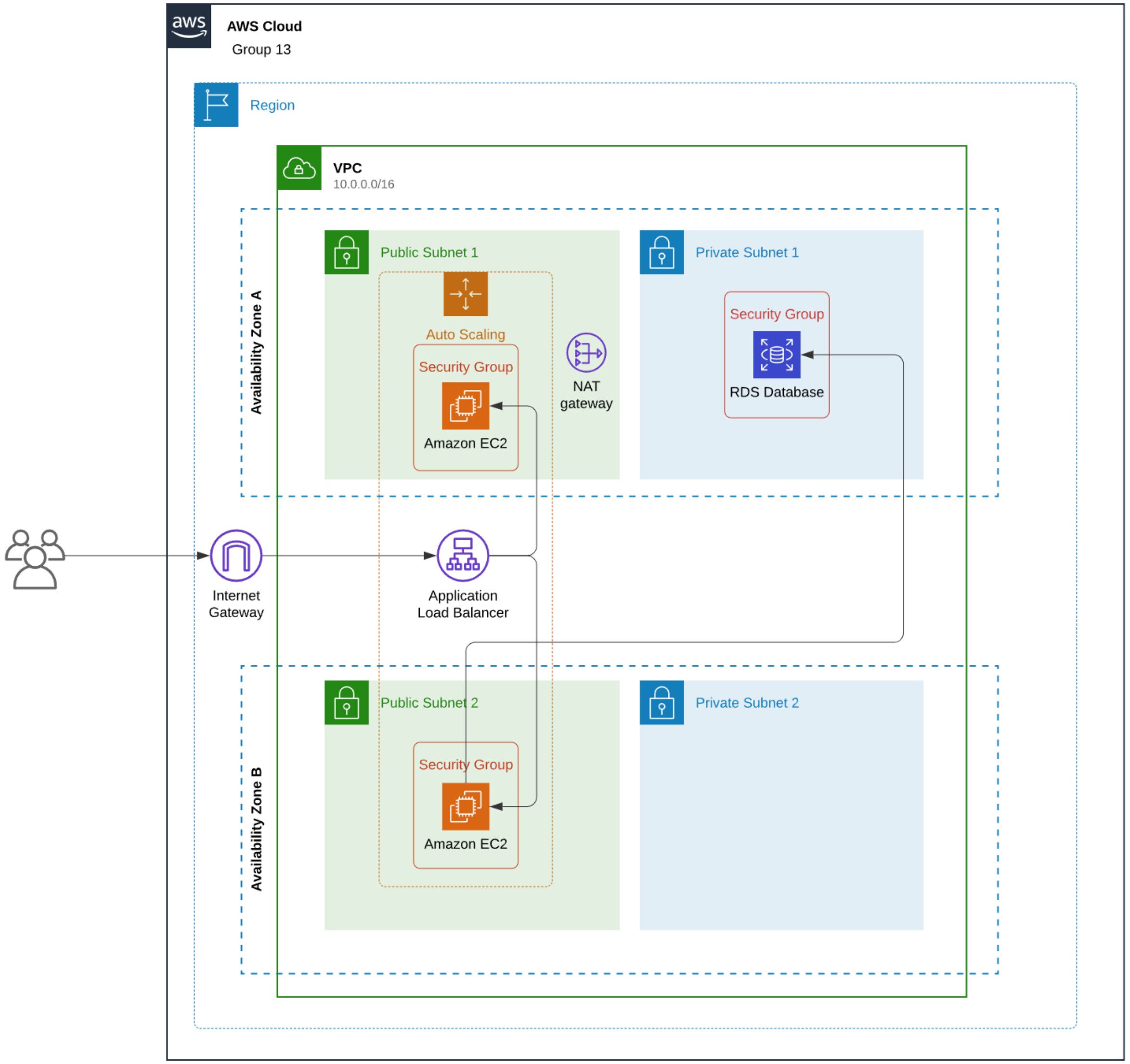


Figure 1. Architecture Diagram

Demo

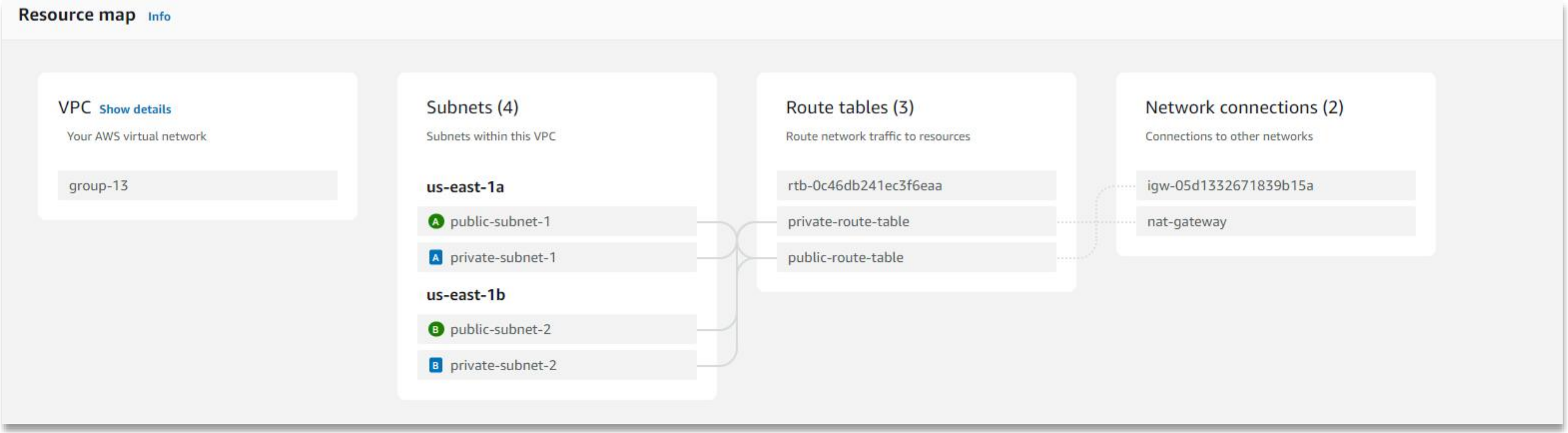
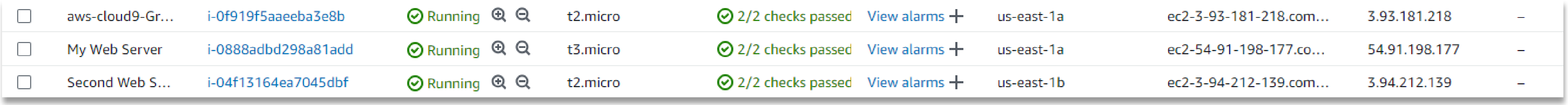


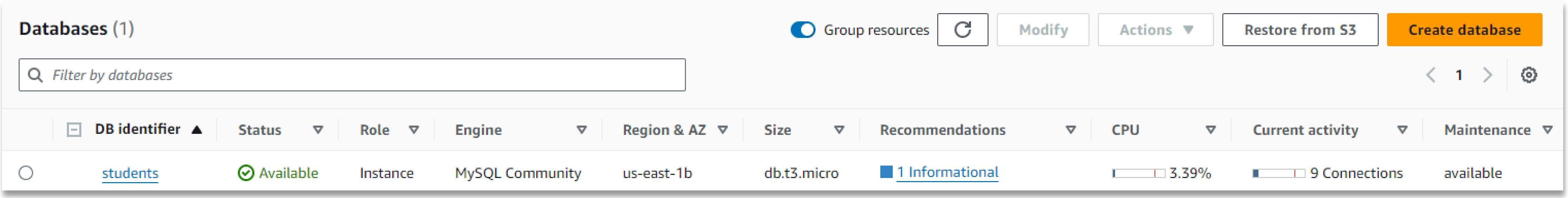
Figure 2. VPC Setup

Demo



□	aws-cloud9-Gr...	i-0f919f5aaeeba3e8b	✓ Running	t2.micro	✓ 2/2 checks passed	View alarms +	us-east-1a	ec2-3-93-181-218.com...	3.93.181.218	–
□	My Web Server	i-0888adbd298a81add	✓ Running	t3.micro	✓ 2/2 checks passed	View alarms +	us-east-1a	ec2-54-91-198-177.co...	54.91.198.177	–
□	Second Web S...	i-04f13164ea7045dbf	✓ Running	t2.micro	✓ 2/2 checks passed	View alarms +	us-east-1b	ec2-3-94-212-139.com...	3.94.212.139	–

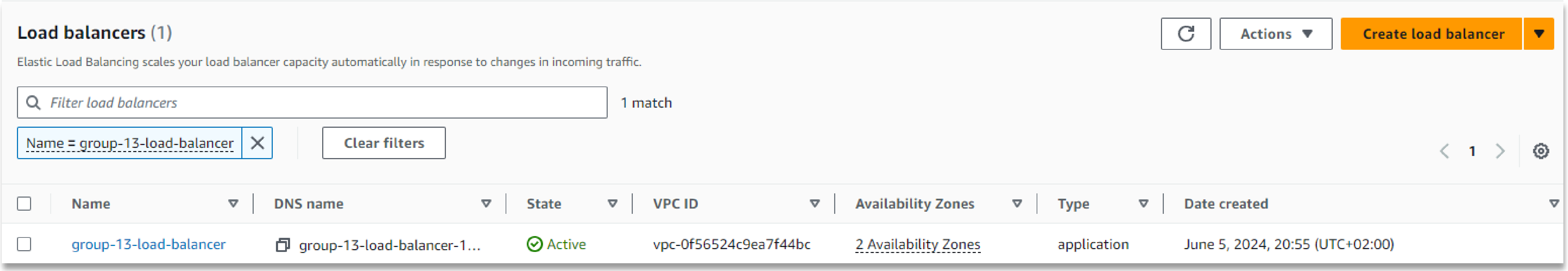
Figure 3. EC2 Instances



	DB identifier ▲	Status	Role	Engine	Region & AZ	Size	Recommendations	CPU	Current activity	Maintenance
○	students	✓ Available	Instance	MySQL Community	us-east-1b	db.t3.micro	1 Informational	3.39%	9 Connections	available

Figure 4. RDS Database

Demo



Load balancers (1)

Elastic Load Balancing scales your load balancer capacity automatically in response to changes in incoming traffic.

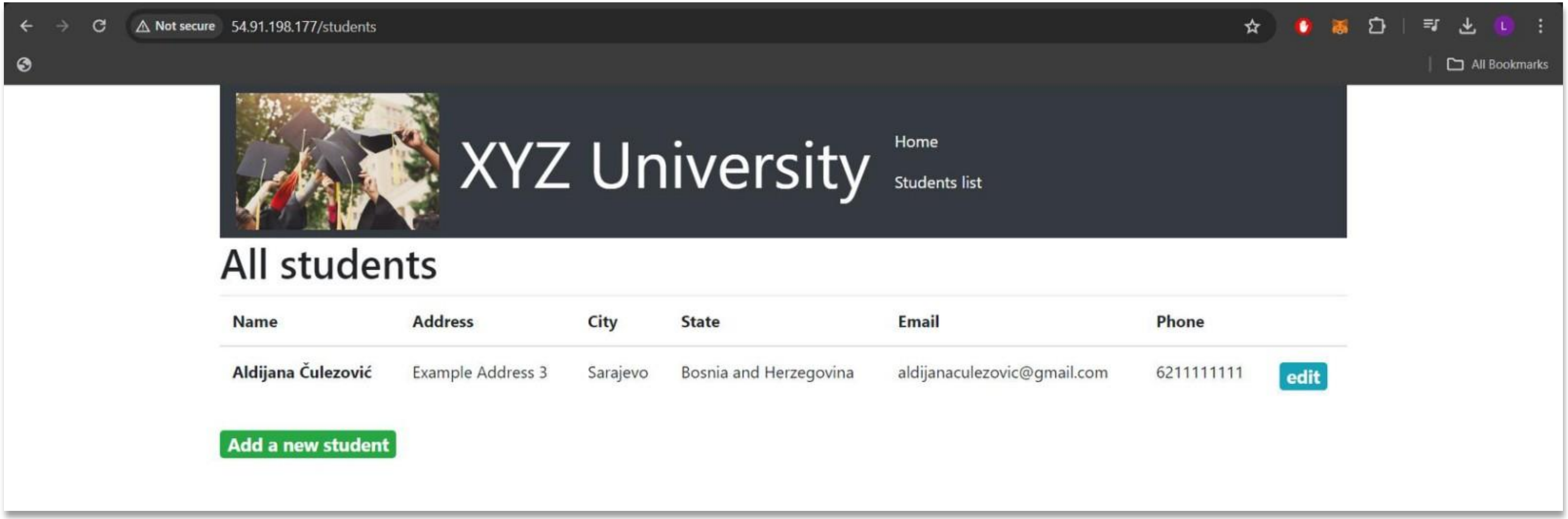
1 match

Name = group-13-load-balancer

□	Name	DNS name	State	VPC ID	Availability Zones	Type	Date created
□	group-13-load-balancer	group-13-load-balancer-1...	✓ Active	vpc-0f56524c9ea7f44bc	2 Availability Zones	application	June 5, 2024, 20:55 (UTC+02:00)

Figure 5. Load Balancer

Demo

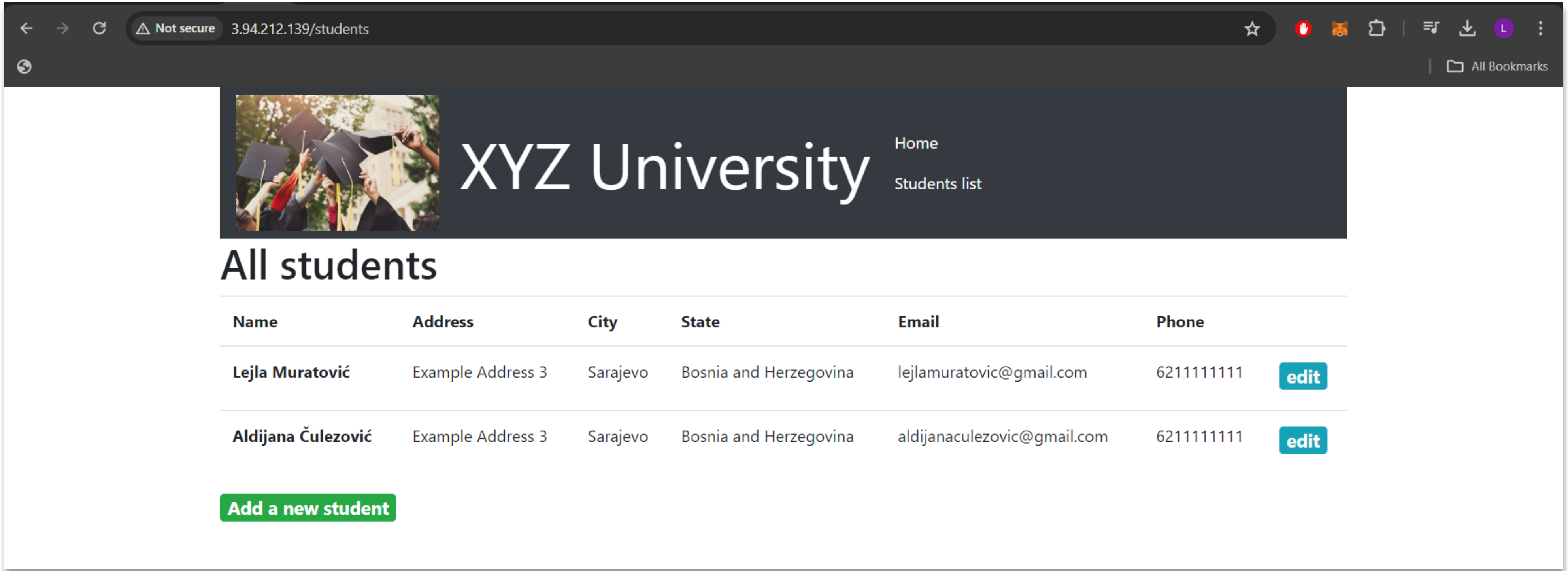


All students

Name	Address	City	State	Email	Phone	
Aldijana Čulezović	Example Address 3	Sarajevo	Bosnia and Herzegovina	aldijanaculezovic@gmail.com	6211111111	edit

Figure 6. First EC2 Instance

Demo



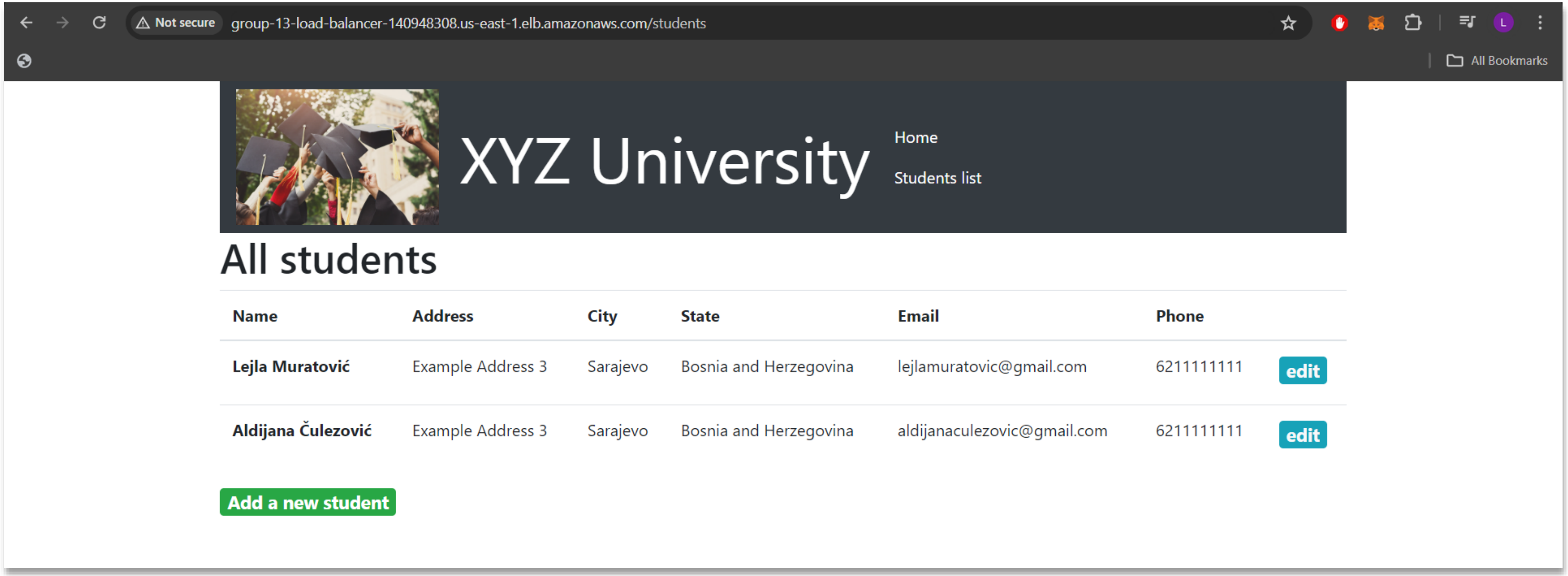
XYZ University

All students

Name	Address	City	State	Email	Phone	
Lejla Muratović	Example Address 3	Sarajevo	Bosnia and Herzegovina	lejlamuratovic@gmail.com	6211111111	edit
Aldijana Čulezović	Example Address 3	Sarajevo	Bosnia and Herzegovina	aldijanaculezovic@gmail.com	6211111111	edit

Figure 7. Second EC2 Instance

Demo



Not secure group-13-load-balancer-140948308.us-east-1.elb.amazonaws.com/students

All students

Name	Address	City	State	Email	Phone	
Lejla Muratović	Example Address 3	Sarajevo	Bosnia and Herzegovina	lejlamuratovic@gmail.com	6211111111	edit
Aldijana Čulezović	Example Address 3	Sarajevo	Bosnia and Herzegovina	aldijanaculezovic@gmail.com	6211111111	edit

Figure 8. Load Balancer

Lessons learned

Challenges & Resources

- At first, we experienced issues with database migration as it was difficult to understand the shell commands at first, but after following the AWS Academy Cloud Architecting – Lab: Migrating a Database to Amazon RDS we successfuly completed the task and database was migrated into the EC2 Instance.
- We also had problems understanding the steps of Load Balancer and how it actually works. We followed instructions from AWS Academy Cloud Architecting – Lab: Creating a Highly Available Environment, which helped us understand the concept and how to implement it.

New skills

- We gained knowledge on how to use AWS services like EC2, RDS, ELB, and Auto Scaling to design and build a highly available, scalable, and secure online application. We acquired skills in setting up security groups and virtual private clouds to improve application security and guarantee data integrity. During this project, we have learned that a load balancer efficiently distributes incoming network traffic across multiple servers to ensure no single server bears too much demand.

Next steps

- Create database backups and further scale the application.
- Secure the application by implementing HTTPS to encrypt data in transit.
- Utilize CloudWatch and CloudTrail for detailed monitoring and logging.

Questions?

THANK YOU!