THE GEORGE WASHINGTON UNIVERSITY

WASHINGTON, DC

1. Course Overview

CSCI 2541 Database Systems & Team Projects

~~Wood~~ Gabe

Slides adapted from ones by Prof. Bhagi Narahari & Wood

CS 2541W: Database Systems & Team Projects

Spring 2026

Professor: Gabe Parmer

TAs: Emil Abbasov, Bella Dayrit, Sydney Berritt
LA/Support Staff: Adita Arjun, Zeyuan Yang

https://gwu-cs-db-s26.github.io

Gabe

(random facts)

I’ve taught:

- 1111
- Systems Programming,
- Operating Systems,
- Senior Design,
- IoT,
- Advanced OS

I like: Operating Systems, Embedded Systems, Programming languages, Board games, Netrunner, Coding & Learning

Course Staff

Grad TAs: Labs, office hours, project mentoring, grading

- Emil

Undergraduate team: Labs, office hours, project mentoring

- Bella, Sydney, Aditya

Inappropriate interactions:

- DMs
- Disrespect (one warning, then lose a HW’s credit)
- Asking about grades

Course Requirements & Logistics…

Course Objectives

Relational database theory and design

- Concepts of data storage and retrieval

Fluency in SQL and web front- and back-end development

- Working with relational database systems: MySQL
- Python to develop DB connected apps

Software integration experience and team S/W development experience

- Design and deploy a large database application
- Full stack (web) development

Brief introduction to NoSQL/analytics/ML DB models

Table Chat

This class is about 1. databases, 2. full-stack development, and 3. teamwork.

What are each of these?

Are they important? Why?

Course Requirements: Grading

Exam (midterm): 22.5%

- Based on lectures and labs. Around Week 6-7

Homework, Lab Assignments: 25%

- Programming and written homework
- In-Lab/Class exercises given out during class and equivalent to a “quiz”

Team Project: 37.5%

- Phase 1 (15%) + Phase 2 (22.5%)
- No final exam BUT final project demos are required
- To pass project, your demos have to work…NO broken websites!

Team Contribution: 10%

- How well do you work with others in labs and projects

Engagement: 5%

- Participation in class or online with a variety of bonus opportunities
- Usually limited to 2 points per week; Need 15 points for full credit

Grades curved (and scaled as percentage of highest score in class)

Approximate grading method after curving and scaling

A- to A: 90-100%
B- to B+ : 80 to < 90
C- to C+: 70 to < 80
D-: >60

Late Policy

All **group** assignments must be submitted by the deadline; no late work accepted

- Applies to most labs and team projects
- Find a way to get it done; partners will report if work was evenly divided and you will lose points if you did not do your share

For individual assignments you may ask for an extension if needed

- I give the first extension for any reason...
- But you must recognize that submitting late places an extra burden on the staff and disrupts our ability to give feedback to other students
- See the form on the webpage

The Project

A significant part of your grade for the course is a large database systems project

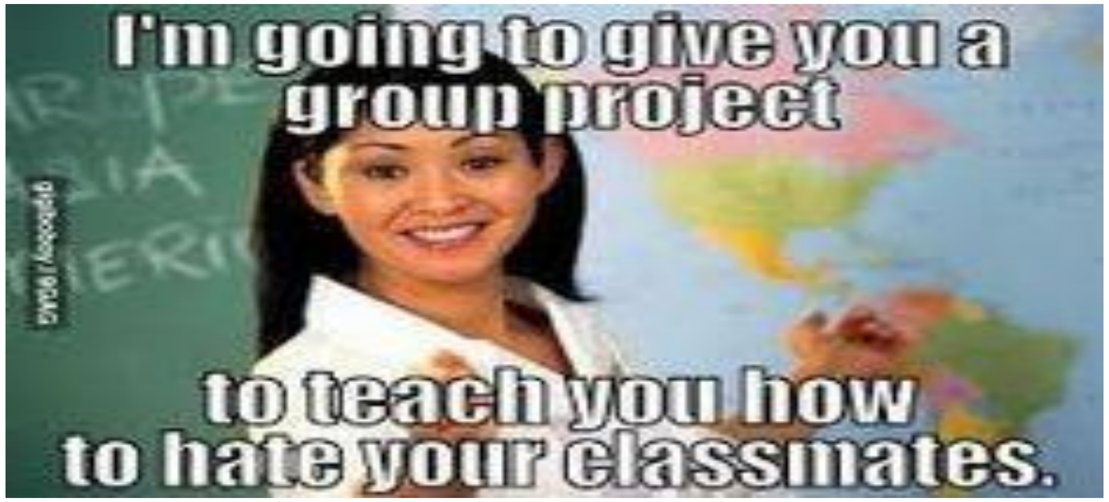
In the project you will design & implement a database system

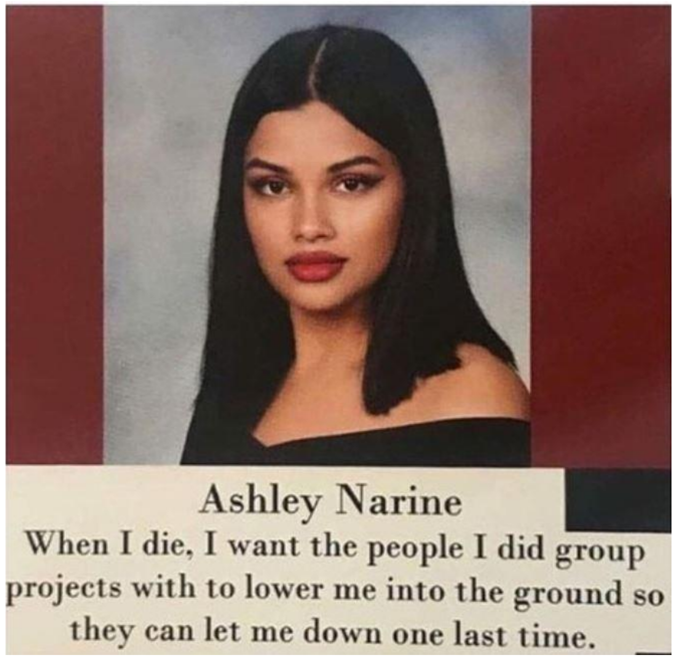
- Full stack development:
  - Front End (HTML/CSS & optional Javascript)
  - Application server – in Python
  - DBMS backend – MySQL
- All the above are useful (high demand) skills
  - Note that limited background will be given on web programming (seek help from TAs)

The project will involve working in teams of 2 to 4

- Larger teams must develop projects with more features

Why do we have team projects


I'm going to give you a group project
to teach you how to hate your classmates.


Ashley Narine

When I die, I want the people I did group projects with to lower me into the ground so they can let me down one last time.

You make half of the assignment and I'll make the other half and then we'll join them together

Real World: Teamwork and S/W development in teams is the default!

- Communication
- Collaboration
- Conflict resolution
- Using tools to enable collaborative SW dev

Team Project: Requirements & Expectations

Project broken into 2 phases:

- **Phase 1**: teams to build an application assigned to your team
- **Phase 2**: Work in new teams to integrate different applications and produce the final project
  - Take what you built in Phase 1 and integrate with systems built by others….
  - This requires **integration** and **NOT redesign**

You HAVE to deliver a working project for credit

Teamwork Assessment…part of your grade!

You have to work in teams

- Each team member required to ‘produce’ equitable
- Teamwork will be assessed…
  - Not all team members may get the same grade on the project!
  - You must bring teamwork issues to attention of the instructor
  - If you do not contribute your fair portion of Phase 1, then you may be required to complete Phase 2 **on your own**

The second half of the course will have one session (lecture or a lab) dedicated to teamwork check-ins each week

- Instruction team will meet with each team, and assess if the weekly deliverables are being met by each team member

If you cannot commit time each week to working on the team project then please drop the course!

Lab Sections: treated as one lab section

Lab sections will cover:

- HTML/CSS
- Python and Flask web framework
- SQLite/MySQL
- Intro to a NoSQL DB - MongoDB
- Clarifications on Programming Assignments

In-class assignments in some weeks

- Will be graded, with async option

Academic Integrity Policy

No collaboration on homework/programming assignments

- Including external resources, tutors, AI, online help
- Okay to clarify questions for a classmate
- **Not okay** to share solutions or solution code

No collaboration between teams on team projects

- within team each team member must have clear role -- i.e., clearly partitioned tasks for each team member

No use of LLMs or coding assistants

Cheating or not doing the work is the only way to fail the class. Cheating is the easiest way to fail.

Your Spring Semester

Foundations

Makes you **think in new ways** and understand the underlying **principles** and **algorithms** of complex software

Architecture

Teaches you more about how CPUs work, and how to interact with the hardware

Database Systems

Practical experience with web and DB programming, and working in a team on a substantial project

What is a Database System?

A Database is a collection of related data

- Typically **carefully structured** with well defined relations
- Models **entities and relations**
- **Persistent** store for data

A Database Management System (**DBMS**) is the software system to store/retrieve/manage the database.

- Provides an interface over the database
- Examples: Oracle, MySQL, MongoDB, Postgres, Dynamo...

A Database System = DBMS + Data + Application

- In this course, we will use MySQL + Python (Flask web app framework)

What is this course?

Database systems design and implementation

- Theory of relational database design and query languages
  - Relational Model, Relational algebra, SQL
- Application development using Relational DBMS (MySQL), with Python web apps

Intro to database models for unstructured data (Big data)

- Overview of NoSQL database models, analytics, and ML DBS

What is this course?

Database systems design and implementation

- ○ Theory of relational database design and query languages
  - ● Relational Model, Relational algebra, SQL
- ○ Application development using Relational DBMS (MySQL), with Python web apps

Intro to database models for unstructured data (Big data)

- ○ Overview of NoSQL database models, analytics, and ML DBS

Database System Project: Full stack development

Teamwork – SW development in teams

- ○ Project (SW) integration

*Course is not just about Database design – you have to learn and participate in the other course objectives.

What is this course?

Week 1: Intro to DBs, HTML, CSS

- Also: Learn Python **on your own**!

Weeks 2-3: Relational Database Model and SQL

Weeks 4-5: Good Database Design Practices

Week 6: EXAM!

Week 7: Pivot to project

Weeks 8+: Team project time!

- Phase 1: Build a web app
- Phase 2: Integrate your web apps

What is this course?

- Context and knowledge:
  Databases as the way we store/access data
- Practical skills:
  Web programming & internet-connected services
- “Soft” skills:
  Teamwork & communication is hard!

Table Chat

What is **HTML/CSS**?
What is it used for?

Have you heard of **Python**?
What is Python used for?

What makes these technologies
interesting to you (if they do)?

Intro to Databases & Database Management Systems

Databases in the Real-World

Databases are everywhere in the real-world even though you do not often interact with the DBMS directly.

- ~$50 billion annual industry

Examples:

- Retailers manage their products and sales using a database.
  - Wal-Mart has one of the largest databases in the world ~40 Petabytes !
- Online web sites such as Amazon, eBay, etc..
- Social media sites: Facebook adds >500 Terabytes of new data per day!
- The university maintains all your registration information in a database.
- Mobile apps need to store your local data somewhere
- **Every** mobile app/web browser uses Sqlite

DBMS Examples

There are many different Database Management Systems

- In this class we will (mostly) use SQlite & MySQL

Have you heard of any other database software platforms?

An Example: RestaurantDB

Your aunt and uncle want you to help track the top customers at their restaurant

- When are upcoming reservations?
- Which customer visits the most often?
- Who spends the most money?
- How to reward customers who order 10+ dishes?
- What is the most popular dish?
- What is the most popular dish on Tuesdays?

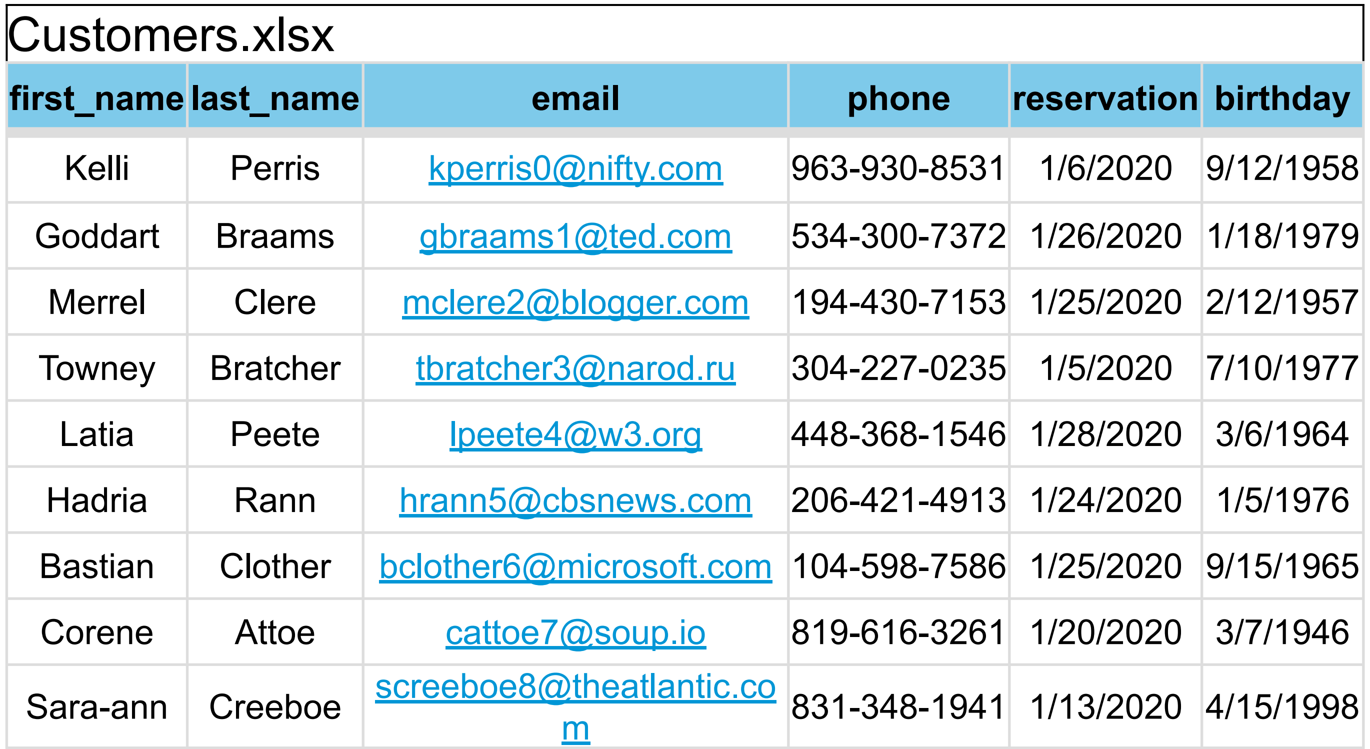
Why not just use Excel?

A spreadsheet can easily store data

- Use columns to structure information
- Enter a new row for each restaurant customer
- Can use formulas to calculate answers to some queries or sort/filter table

But…

Why will Excel have problems with this? What will be difficult?



Customers.xlsx

first_name	last_name	email	phone	reservation	birthday
Kelli	Perris	kperris0@nifty.com	963-930-8531	1/6/2020	9/12/1958
Goddart	Braams	gbraams1@ted.com	534-300-7372	1/26/2020	1/18/1979
Merrel	Clere	mclere2@blogger.com	194-430-7153	1/25/2020	2/12/1957
Towney	Bratcher	tbratcher3@narod.ru	304-227-0235	1/5/2020	7/10/1977
Latia	Peete	lpeete4@w3.org	448-368-1546	1/28/2020	3/6/1964
Hadria	Rann	hrann5@cbsnews.com	206-421-4913	1/24/2020	1/5/1976
Bastian	Clother	bclother6@microsoft.com	104-598-7586	1/25/2020	9/15/1965
Corene	Attoe	cattoe7@soup.io	819-616-3261	1/20/2020	3/7/1946
Sara-ann	Creeboe	screeboe8@theatlantic.com	831-348-1941	1/13/2020	4/15/1998

Why not just use Excel?

A spreadsheet can easily store data

- Use columns to structure information
- Enter a new row for each restaurant customer
- Can use formulas to calculate answers to some queries or sort/filter table

But…

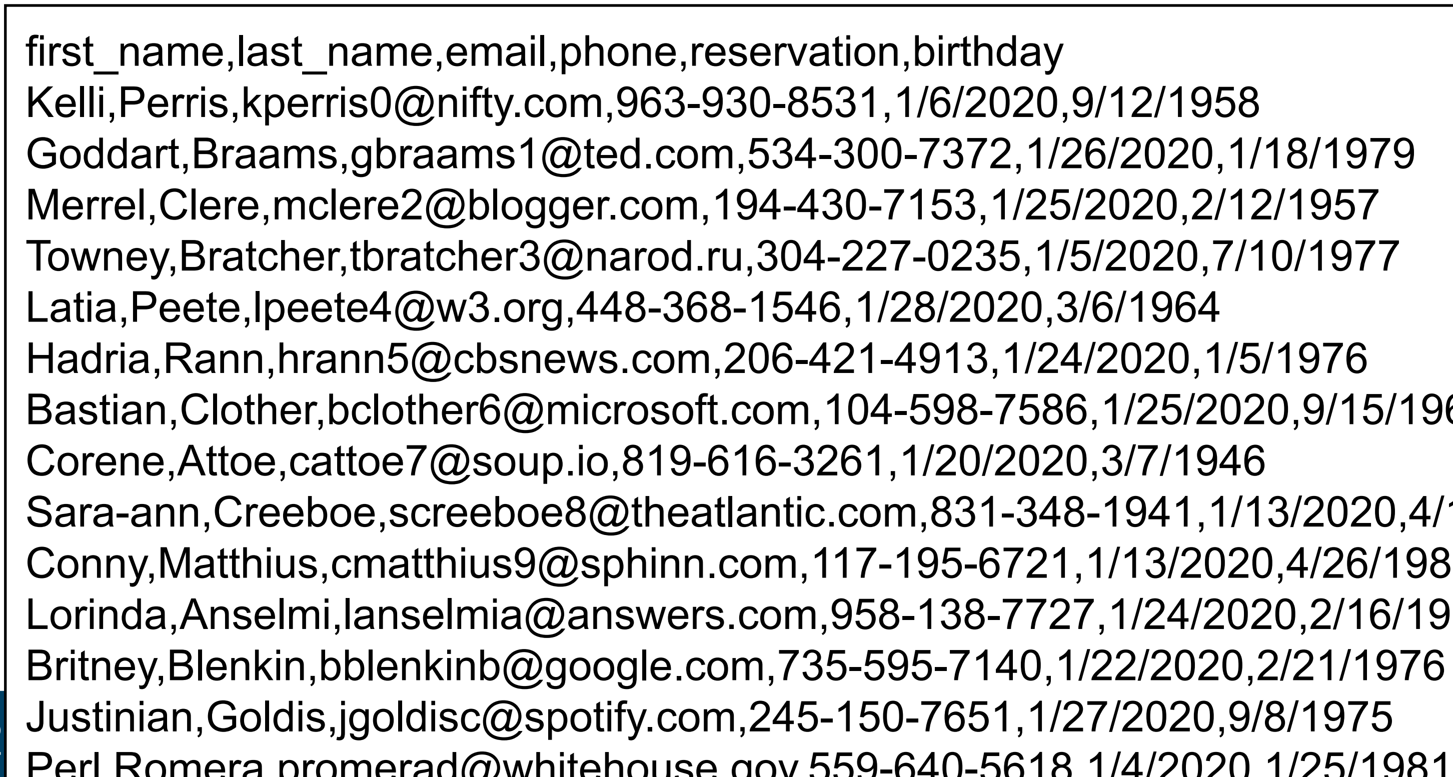
- Some calculations may not be easy (or possible) to write as *limited excel formulas* – need more power!
- Sorting/filtering tables to see different results will get *messy*
- Need to be careful about entering data (*no data validation* of format/data type)
- Can *accidentally corrupt old data* since data storage and processing are combined in one place

Why not just use files?

You already know how to read and write data to a file…

- Could store data in Excel, then export to CSV (comma separated value file)
- Program could read file's lines and store into a data structure e.g., a Linked List
- Different functions could calculate answers for different queries

Why will file processing have problems with this? What will be difficult?



```
first_name,last_name,email,phone,reservation,birthday
Kelli,Perris,kperris0@nifty.com,963-930-8531,1/6/2020,9/12/1958
Goddart,Braams,gbraams1@ted.com,534-300-7372,1/26/2020,1/18/1979
Merrel,Clere,mclere2@blogger.com,194-430-7153,1/25/2020,2/12/1957
Towney,Bratcher,tbratcher3@narod.ru,304-227-0235,1/5/2020,7/10/1977
Latia,Peete,lpeete4@w3.org,448-368-1546,1/28/2020,3/6/1964
Hadria,Rann,hrann5@cbsnews.com,206-421-4913,1/24/2020,1/5/1976
Bastian,Clother,bclother6@microsoft.com,104-598-7586,1/25/2020,9/15/196
Corene,Attoe,cattoe7@soup.io,819-616-3261,1/20/2020,3/7/1946
Sara-ann,Creeboe,screeboe8@theatlantic.com,831-348-1941,1/13/2020,4/
Conny,Matthius,cmatthius9@sphinn.com,117-195-6721,1/13/2020,4/26/198
Lorinda,Anselmi,lanselmia@answers.com,958-138-7727,1/24/2020,2/16/19
Britney,Blenkin,bblenkinb@google.com,735-595-7140,1/22/2020,2/21/1976
Justinian,Goldis,jgoldisc@spotify.com,245-150-7651,1/27/2020,9/8/1975
Perl,Romera,promerad@whitehouse.gov,559-640-5618,1/4/2020,1/25/1981
```

Why not just use files?

You already know how to read and write data to a file…

- Could store data in Excel, then export to CSV (comma separated value file)
- Program could read file's lines and store into a data structure e.g., a Linked List
- Different functions could calculate answers for different queries

But…

- File system doesn't know anything about file format used in our program - we need to implement all parsing ourselves
- Calculations are tightly tied to data format - if we change format we need to rewrite all the code
- *Complex query code* in a general purpose language!
- *Redundancy* - replicate query code
- Hard to support multiple *simultaneous users*

Scaling up

What if we have a chain of restaurants?

What becomes more complex?

Can Excel/file parsing work in this environment?

Scaling up

What if we have a chain of restaurants?

Uhoh…

- We need to support **concurrent** updates/reads to the data
- We need to ensure data remains **consistent** despite simultaneous access
- Google Sheets might make it easier for multiple users to make edits, but doesn't **guarantee these properties**!
- Expanding our custom file parser to support network access is a major effort!

So what can we conclude thus far….

Excel is limited in **scale**, **flexibility**, and **power**

File processing is not **portable**, **efficient,** nor **easy**

Need a “database approach” that provides **data independence** from the processing acting on it

Need to support **simultaneous access** while retaining data **integrity**

So how do we specify business rules of the data, relationships within the data, who gets access to what data… **How to organize and manage the data ?**

Course Resources

Course webpage: will have links to syllabus, lecture notes, online resources (and in-class exercises when applicable)

- https://gwu-cs-db-s26.github.io/

Github:

- Labs and HW assignments
- Project submissions and team management

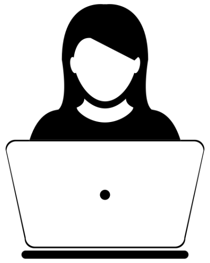
Discord:

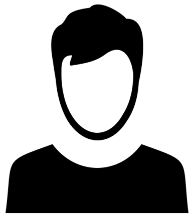
- For class announcements and Q&A
- For team coordination
- Participation points

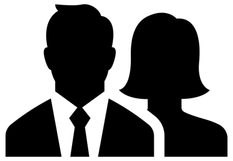
Attributions

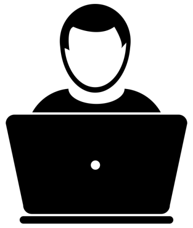
These slides are adapted from materials made by Prof. Bhagi Narahari

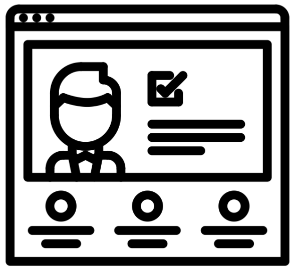
Image attribution:


Created by Wilson Joseph
from Noun Project


Created by Ditta
from Noun Project


Created by Gregor Cresnar
from Noun Project


Created by Wilson Joseph
from Noun Project

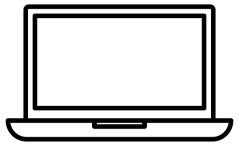

Created by lastspark
from Noun Project

Created by Dawid Sobolewski
from Noun Project

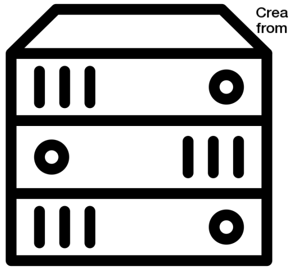
Created by Wilson Joseph
from Noun Project

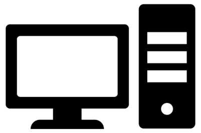
Created by Saifurrijal
from Noun Project

Created by ainul muttaqin
from Noun Project


Created by ainul muttaqin
from Noun Project

Created by ainul muttaqin
from Noun Project


Created by Srinivas Agra
from Noun Project


Created by Yazmin Alanis
from Noun Project