

README.md

Workbook Questions for Part A

Q1 Describe the architecture of a typical Flask application 200-300

Flask application usually consists of a simple architecture known as the Model-View-Controller pattern or framework. With this information we can see that it is divided into three parts with their own roles in an application.

The Model

The model holds the data and business logic of the Flask app and usually it would include the data or classes that control or provide the structure of the data. The model would also be the one that actually talks to other databases to retrieve or modify data. Some things that can be used are SQLAlchemy and Flask-SQLAlchemy to interact with databases in Flask applications.

The View

The view is what is presented to the client and is responsible for the requests of the user. As an example this could be what built a website such as the interface by CSS and HTML. After user requests are sent they are then sent to the controller.

The Controller

The controller acts as a middle man between the model and the view what it does is acts as the processor of information. When the request is sent from the view it then processes the information with help from the model and updates the view as needed. The controller in flask is used by routes and assisted/defined by using decorators such as `@app.route("/")`. This mapping is used to link different routes to different functions in Python. The controller can also be used for authentication, authorization, validation and error handling through decorators such as `@jwt_required()`

Flask Documentation: <https://flask.palletsprojects.com/> Pallets Projects:
<https://palletsprojects.com/> Codr Academy Slides and Information through ntos

Q2 Identify a database management system (DBMS) commonly used in web applications (including Flask) and discuss the pros and cons of this database 150-250

A DBMS that is widely used in web applications including Flask is PostgreSQL which has pros and cons.

Pros

The reliability of PostgreSQL is quite well known and ensures that it is a reliable choice for web applications. It has a wide array of features that can be used for simple and complex queries with a plus of being agnostic in regards to programming languages meaning that it is widely supported and has a large library of usable extensions and support.

If built well PostgreSQL is able to handle applications and datasets that are scaling or already large quite well due to its optimisation and caching making it quite an efficient workhorse. In use where multiple transactions need to complete, the data can be simultaneously completed without interfering with each other ensuring data integrity as well.

Cons

While PostgreSQL is known for its variety of features and flexibility this can also be a con because it can be a bit more complex to set up if only a simpler DBMS can be used instead. Otherwise instead of efficiency it might work slower and will need to be rebuilt later on for scalability. This can also cause memory issues when scaled up because PostgreSQL can require careful tuning to ensure that the memory usage is done correctly in the first place.

PostgreSQL: a closer look at the object-relational database management system
<https://www.ionos.com/digitalguide/server/know-how/postgresql/>

What is PostgreSQL? <https://www.guru99.com/introduction-postgresql.html>

ACME Corporation is very big on project management, documentation and process. This will be a key metric in their decision to award the project. The following set of questions relate to this RfQ-requirement.

Q3 Discuss the implementation of Agile project management methodology 200-300

Agile project management is a methodology that invites a collaborative effort that focuses on flexibility and customer satisfaction. The focus is to create small frequent changes with continuous feedback on these changes which then iterate and iterate until a project is successfully completed. It promotes a close relationship between stakeholders and teams allowing a quick response to change with a mindset focused on adaptability.

An example of a framework is called Scrum which is a method that divides the project in "sprints", typically there are daily meetings for progress updates and a review done at each end of the sprint with reviews of what happened. This type of methodology promotes continuous progress and improvement whilst also allowing adaptation to any feedback with daily standups and reviews of sprints.

Between the team members and the stakeholders, this style of management is possible because of what is known as the "project backlog", which is a set of features, needs, requirements that are and are being prioritised. This list due to the agile methodology is being constantly updated with what is needed. By doing this in theory software can be delivered earlier and more frequently/

At the heart of it all, Agile project management methodologies is a flexible and adaptive style of management that focuses on assisting with teams to consistently adapt to any changing requirements that are needed or removed through feedback and customer value. It allows requirements to evolve as needed and delivers successful projects through a style of iteration and collaboration.

What Is Agile Methodology in Project Management? <https://www.wrike.com/project-management-guide/faq/what-is-agile-methodology-in-project-management/>

Beginner's Guide to Agile Project Management <https://business.adobe.com/blog/basics/agile>

Q4 Provide an overview and description of a standard source control workflow 100-200

A standard source control workflow used is called the branch workflow. This type of workflow a new feature or task is separated from the others and made into its own "branch" and then once completed will merge it back into the main database of code. This workflow helps to keep the main database of code clean and stable as many developers may be working on different projects all at once as well as being able to assist each other with things such as pull requests.

In the branch workflow it usually involves the following steps:

1. Branch Creation: New branch is made for feature or task
2. Development: Changes are made and committed to their code to test

3. Pull Request: Feature is completed and is sent of to check and review changes
4. Code Review: Other members check the code before signing off
5. Merge: If code is apporved the branch is merged back into branch for a enw feature in the codebase
6. Testing: Tests are compelted to ensure that this change is not causing/has not caused any issues.
7. Deployment: Code is sent into production(sent to be used by clients)

Git Feature Branch Workflow <https://www.atlassian.com/git/tutorials/comparing-workflows/feature-branch-workflow>

Q5 Provide an overview and description of a standard software testing process (e.g. manual testing) 100-200

The standard software testing process is a systematic approach that has mutliple sages to identity, assess and check the software follows the requirements set out.

In general the steps taken for testing is usually:

1. Planning: The process starts with planning tests that can check if the software meets the requirements with test cases, testables and scope.
2. Design: After the plan is completed tests are made based on the plans such as expected outputs, inputs, and how to complete a test.
3. Implementation: Test designs are implemented watching the requirements vs the actual results which can include defects or bugs.
4. Error Handling: Any bugs are reported and set as a priority of fixing which are then fixed.
5. Completion and Reporting: Test results are documented and bugs are all adressed.

Through following a standard software testing process software development teams are able to find and assess defects, errors, bugs and unexpected results in software functionality. They are also able to determine if the software meets the minimum requirements and therefore also should be able to provide a high qulaity product that satisfies client expectations.

A Brief Guide to Software Testing - Standards and Processes <https://www.specbee.com/blogs/software-testing-standards-and-processes>

Fundamentals of Software Testing: Concepts and Process <https://www.simplilearn.com/tutorials/devops-tutorial/fundamentals-of-software-testing>

Having suffered several cyber attacks in the past and resultant remedial audits ACME

Corporation takes compliance, security and privacy very seriously. The following set of questions relate to this RfQ-requirement.

Q6 Discuss and analyse requirements related to information system security and how they relate to the project 100-200

In regards to the project since it is for a product line that will be sold to customers the minimum requirements of security for this would be to protect;

1. Customer data
2. Confidentiality
3. System integrity

It is crucial that data privacy is secure especially in compliance with the data protection regulations. This would involve appropriate security measures such as secure data storage, encryption and access controls as data breaches and unauthorised access needs to be protected from. With this a secure way to identify a user must be in place whether it be for user or admin for example requiring two factor authentication or password policies or lockouts.

HTTPS should be employed if possible as it's a secure communication protocol to ensure any data transferred is done properly and securely, this can be enhanced using encryption technology such as the SSL security protocol. Furthermore since this is an application for a product line there needs to be a secure payment method in accordance with the global standard of the Payment Card Industry Data Security Standard.

Q7 Discuss common methods of protecting information and data and how you would apply them to the project 100-200

Following from the above mentioned information some common methods of protecting information and data are:

1. Encryption: Encryption would be very important in order to protect the very basic needs of the application because only those with the encryption key are able to decipher it. This would help protect data confidentiality with client, data, company and customers.
2. Access Controls: Through having access control options this makes it so that only those who have the correct access are able to access different information, whether it be publicly available or sensitive private data that should only be available to those with the appropriate access control. A simple example of this being used are customers logging into the application and requesting them to have multi factor authentication or a strong

password that is not easily accessible.

3. Regular Updates: A basic implementation to protect information and data is to simply keep up to date with the latest versions and protections needed to protect against vulnerabilities and exploits. A regular patch will allow the software used to keep out any found from the previous version.

11 Web Application Security Best Practices You Need to Know <https://www.mobindustry.net/blog/11-web-application-security-best-practices-you-need-to-know/>

10 Web Application Security Best Practices to Secure Your Data <https://www.dailyrazor.com/blog/10-web-application-security-best-practices-to-secure-your-data/>

6 Essential Data Protection Methods <https://gdprinformers.com/gdpr-articles/6-essential-data-protection-methods>

Q8 Research what your legal obligations are in relation to handling user data and how they can be met for the project 100-200

Two major legal obligations in relation to handling user data and information in Australia are the Privacy Act 1988 and the Notifiable Data Breaches Scheme (Under the Privacy Act). These policies stipulate that in the first place within the private sector with organisations who make a certain threshold must comply to having:

1. A privacy policy
2. Obtain consent for data collections
3. Take reasonable measures to protect information
4. The information collected can only be retrieved if it is necessary for their work and eventually remove it if no longer needed
5. Must notify customers as well as the Office of the Australian Commissioner in the event of a data breach and react promptly to the breach

In regards to the project these needs can be met by following the steps that were mentioned above in the previous questions where things such as encryption, multi factor authentication, access controls, and protocols can all be used and implemented to meet reasonable measures to protect information. However there must also be constant checks to know if data is needed or not for retrieval or deletion.

Retention and deletion of personal information collected during COVID-19
<https://www.oaic.gov.au/privacy/privacy-guidance-for-organisations-and-government-agencies/covid-19/retention-and-deletion-of-personal-information-collected-during-covid-19>

Use and disclosure of personal information <https://www.oaic.gov.au/privacy/your-privacy->

[rights/your-personal-information/use-and-disclosure-of-personal-information](#)

Collection of personal information <https://www.oaic.gov.au/privacy/your-privacy-rights/your-personal-information/collection-of-personal-information>

Privacy <https://www.ag.gov.au/rights-and-protections/privacy>

Data Protection Laws and Regulations Australia <https://iclg.com/practice-areas/data-protection-laws-and-regulations/australia>

ACME Corporation has specifically requested the app to be based on a relational database. The next set of questions relate to this RfQ-requirement.

Q9 Describe the structural aspects of the relational database model. Your description should include information about the structure in which data is stored and how relations are represented in that structure. 100-200

The relational database model organises data into either tables or relations which consists of the elements of rows and columns. Aspects of the structure include the structure in which data is stored and how relations appear.

In a relational database model data is stored in tables which typically represent an entity such as people, organisations or customers. Each row of the table typically represents a single record of that entity and each column tends to represent a specific attribute of that entity that is categorised.

Each row also known as Tuples contains a unique identifier for each row called the primary key which uniquely identifies it from other rows and ensures the integrity of the data. The columns also known as attributes have names and a data type that is stored within the column, this type can be anything such as text, booleans or numbers and can also define constraints onto the data stored within them such as (null=False, unique=True or validations)

The relationship part of the database model allows for tables to link to one another. The common types of these are for example one-to-one, many-to-one or one-to-many. These relationships are defined using **foreign keys** which are attributes within one table that relate to a primary key in the other. This relationship through keys are able to keep data integrity across multiple tables.

Coder-Academy Slides and Notes

Relational Data Model in DBMS | Database Concepts & Example <https://www.guru99.com/relational-data-model-dbms.html>

Q10 Describe the integrity aspects of the relational database model. Your description should include information about the types of data integrity and how they can be enforced in a relational database. 100-200

In a relation database integrity aspects are used to ensure consistency and reliability of the data within it. This model type provides several ways to consistently implement the integrity within the model.

Key constraints or entity integrity is used as the main attribute that is unique per row. This is known as the primary key and are never null. This is very important for integrity for relational databases as it is used to differentiate between different the rows and ensures that the data is uniquely attached to a different ID protecting the integrity of each row.

Following from this are the use of foreign keys in a relation database, and this is also known as referential integrity constraints. For a relation database a foreign key is used to reference between different tables. The foreign key must exist in the database as a primary key in a different table and that is how they relate to one another. This foreign key usage prevents references from being orphaned or becoming invalid.

Domain integrity relates to the columns of the table within the database, what this does is that it keeps the integrity of the value within the column within the constraints defined.

Coder-Academy Slides and Notes

Relational Data Model in DBMS | Database Concepts & Example <https://www.guru99.com/relational-data-model-dbms.html>

Q11 Describe the manipulative aspects of the relational database model. Your description should include information about the ways in which data is manipulated (added, removed, changed, and retrieved) in a relational database. 100-200

As an example to be more specific the Structured Query Language (SQL) is widely used

language when it comes to using a relational database. SQL provides commands that are able to execute operations on tables and retrieving data from a relational database. The data manipulation however is always controlled by sticking to the rules of the data, constraints and rules of the database.

SQL is able to provide a few examples of how to manipulate data:

1. INSERT: By using the INSERT operator data can be inserted into an existing table using the table name and values for columns in the row that will be made.

```
INSERT INTO players (id, rank, elo, kda) VALUES (201, Radiant, 3011, 1.3)
```

2. UPDATE: Through the use of UPDATE, existing data within the table can be updated and changed when needed, it tends to have conditions that allow a choice to be made on which rows are to be changed as well.

```
UPDATE players SET elo = 2900 WHERE id = 201
```

3. DELETE: The delete operation as the name suggests is used to remove data from the table by specifying the table name and delete based on conditions.

```
DELETE player WHERE id = 201
```

4. SELECT: This operator is used to retrieve data from the database, it is not limited to a single table and involves specifying columns that are wanted and applying conditions for the rows if needed.

```
SELECT kda FROM players WHERE rank = radiant
```

Companies (including ACME Corporation) value previous project experience and case studies. The following set of questions relate to this RfQ-requirement.

Q12 Conduct research into a web application (app) and answer the following parts: 50-100 per part

The app I have chosen is the productivity app called Notion which can be used on the Web as well as an application on windows and apple platforms.

a. List and describe the software used by the app.

Backend Infrastructure: Notion definitely uses a robust backend because it stores a large

amount of data and synchronises this between all user devices that they log into. This infrastructure would include things such as servers, databases and cloud storage.

Frontend: The frontend of Notion is likely built using technologies such as HTML, CSS and Javascript as it allows for the user interface to be responsive and interactive.

DBMS: A database management system is definitely used and a sophisticated is definitely used for a productivity system such as notion. The DBMS is one that allows for schemas to be changed on the go and definitely is a relational type considering how the different tables and pages can all connect with each other.

Text Editor(Rich): Notion being a productivity app that allows users to manipulate the text that they create with styling options must also use a rich text editor whether it be one that is built by Notion for an inhouse solution or something with similar functionality such as editorjs or slate.

Data encryption and Security: Considering user data is quite private and is a sensitive topic, Notion being a software that is all about user personal information must use encryption and security techniques that protect data while in the app as well as during the protocols.

b. Describe the hardware used to host the app.

Servers: Notion as a online web application that stores and uses large amounts of data must use servers to store and handle this information. This can be either a data center or a cloud system.

Network Equipment: As an online web application Notion requires reliable and speedy hardware so that users and the servers are able to communicate with each other. Equipment such as routers, switches and load balancers are used to facilitate data requests and manage network traffic.

Data Storage: Related to servers data storage devices must be used to store user data which can vary from notes, documents, images and other types of information. This storage can be done through drives such as HDDs, SSDs and typically it's a mix of both. There may also be a backup type of devices especially with a web app with so much data hardware components may include things such as backup servers or storage that regularly back up data as well as keep serviability up if there are system failures or crashes.

c. Describe the interaction of technologies within the app

There are several key interactions of technology within Notion.

As a productivity application that is highly flexible, the database technology allows users to

manage storage, retrieval and manipulation of data making it easy to create and manage content within Notion. This is all possible through a MVC which is pertaining to the Web Technologies that allow a user to interact with Notion and the database. The Web technologies give users a GUI to make these changes and the MVC handles the processing of the requests from client to server and updates accordingly.

d. Describe the way data is structured within the app

Notion is structured using a block-based structure. The blocks in notion each have their own ID and each block has a type that determines how it is rendered within the interface, the type also defines how the properties are interpreted within it. In addition to the attributes that the block itself has they each have attributes that show how to interpret the relationship that they have with other blocks.

The data model behind Notion's flexibility <https://www.notion.so/blog/data-model-behind-notion>

e. Identify entities which must be tracked by the app

1. Pages
2. Blocks
3. Databases (User made)
4. Views
5. Tasks / To Do Lists
6. Projects
7. Notes / Documentation
8. Users
9. Files and Attachments

f. Identify the relationships and associations between the entities you have identified in part (e)

1. Pages
 - Pages can have a one-to many relationship with blocks as a page can contain multiple blocks and blocks need a page
 - Pages can have a one to many relationship with users as multiple uses can contribute to a page and they can also have multiple pages
 - Can have zero to many with databases as multiple databases can connect with multiple pages and can exist without a database
2. Blocks
 - Can have a many to one relationship with pages as many blocks can only be connected to one page and need a page to exist

- Can have a zero to many relationship with files and attachments as one block can have multiple attachments and can exist without a file
- Can have a many to zero relationship with projects as one block can have multiple projects referenced and can exist without one
- Can have a many to zero relationship with tasks as many blocks can reference a single task and can exist without one

3. Databases

- Can have a one to many relationship with views as a single database can have multiple views and cannot exist without one
- One to many relationship with users as multiple users can have multiple databases and cannot exist without one
- Can have one to many with pages as multiple databases can connect with multiple pages and cannot exist without one

4. Views

- Many to one relationship with databases as multiple views can be used for a single database and cannot exist without one

5. Tasks/to do list

- Zero to many relationship with users multiple users can be assigned to a single task and can exist without one
- Zero to many relationship with projects because a project can have multiple or none tasks assigned and can exist without one

6. Projects

- One to many relationship with tasks because a project can have many to none tasks assigned and can exist without one
- Zero to many relationship with page because a project can exist on multiple pages but a page does not need a project

7. Notes / Documentation

- One to Many relationship with users as multiple or one users can be assigned access

8. Users

- Zero to many relationship with pages as a user can have no pages (empty account)
- Zero to many relationship with tasks as users can be assigned none or many tasks
- Zero to many relationship with notes or documentation as a user can have no notes or documentation or more assigned
- One to many relationship with db

9. Files and Attachments

- one to many relationship with blocks as multiple blocks can be assigned access to files

g. Design a schema using an Entity Relationship Diagram (ERD) appropriate for the database of this website (assuming a relational database model)

