What are some benefits of no-code application development platforms? Select two.

Easier to install and maintain than other development platforms

Improved security and compliance

Faster time to deployment

Testing is not required

Zero IT backlog

3.

Which one of the following statements regarding no-code application development is correct?

No-code application development is the process of developing apps without writing any code.

No-code apps can only run on mobile devices like phones or tablets.

You need some experience writing code even when using no-code app development platforms.

Apps developed using no-code application development platforms do not contain any code.

4.

What are some of AppSheet's capabilities? Select two.

Using AppSheet, you can create and use forms to capture user input data.

AppSheet can be used to stream live video and manage events.

AppSheet enables you to create apps with various types of views including tables, galleries, charts, and maps.
All questions must be answere

ASSISTANT

1. A citizen developer is: **A line-of-business worker in an organization who uses no-code application development tools to develop apps for business functions.**

2. Some benefits of no-code application development platforms include:
   - **Faster time to deployment**
   - **Improved security and compliance**

3. The correct statement regarding no-code application development is: **No-code application development is the process of developing apps without writing any code.**

4. Some of AppSheet's capabilities are:
   - **Using AppSheet, you can create and use forms to capture user input data.**
   - **AppSheet enables you to create apps with various types of views including tables, galleries, charts, and maps.**

USER

ng score: 75%

1.

AppSheet lets you connect spreadsheets as data sources for use with your app. What are two best practices for using spreadsheets with AppSheet?

Ensure that the data type format is the same for each cell in a column.

Name your spreadsheet columns appropriately using a header row.

Limit the number of rows and columns in the spreadsheet.

When you add data to the spreadsheet, it is better to add additional columns so that the data is oriented horizontally.

Do not use formulas in spreadsheets.

2.

As you build and customize your app, which is the quickest and easiest way to test it?

Use the AppSheet-provided browser link to test the app in a separate tab or window.

Use the install link to download and run the app on your device.

Use the live app preview feature in the AppSheet editor.

Share the app with end users so that they can test it.

3.

How can you start creating an app in AppSheet? (Select two methods)

Copy an existing template.

Develop a mobile app outside of AppSheet and deploy it on the AppSheet platform to add additional features.

Write code to develop a web application and upload the code to AppSheet.

Start by configuring a data source for the app.

Copy a script file that contains some code.

4.

After connecting a data source to your app, what is one of the first steps you should take to build the app?

Create actions and behaviors in the app.

Customize the look and feel of the app using colors and branding.

Define the column types of the data and how the data will be used by the app.

Create views so app users can interact with the app and the data.

ASSISTANT

1. Two best practices for using spreadsheets with AppSheet are:
   - **Ensure that the data type format is the same for each cell in a column.**
   - **Name your spreadsheet columns appropriately using a header row.**

2. The quickest and easiest way to test your app as you build and customize it is to **use the live app preview feature in the AppSheet editor**.

3. Two methods to start creating an app in AppSheet are:
   - **Start by configuring a data source for the app.**
   - **Copy an existing template.**

4. After connecting a data source to your app, one of the first steps you should take to build the app is to **define the column types of the data and how the data will be used by the app**.

USER

give me a b c or d answer ing score: 75%

1.

What are some characteristics of AppSheet table references? (Select 2)

Only the app creator can create references when building an app.

A reference is a relationship between related tables that use a column of type Ref in one table that contains a value from a key column of a related row in the other table.

A reference enables the app to easily retrieve information from a related row in another table.

A table that is used in AppSheet must contain at least one reference.

References are unidirectional, so an app can only traverse data from the table that contains the Ref column to the referenced table.

2.

What types of data sources are supported by AppSheet? (Select 2)

Spreadsheets

IoT (Internet-of-things) devices

Databases

3.

Which statements about the key column of a table that is used in AppSheet are correct? (Select 2)

It is acceptable for two or more rows in a table to have the same value in the key column. (duplicate keys)

A key column is always optional in a table.

The value in a key column uniquely identifies each row in the table.

Any table used in AppSheet must define a key column.

Values that are stored in a key column of a table must be numeric.

4.

What is a virtual column in AppSheet?

It is a column that must have a type explicitly set when the column is created.

It is a column that can only be added by the system and not by the app creator.

It is a column that is usually populated using an app formula.

It is a temporary column that is added to the underlying data source for use by AppSheet.

ASSISTANT

1. **B. A reference is a relationship between related tables that use a column of type Ref in one table that contains a value from a key column of a related row in the other table.**

   C. A reference enables the app to easily retrieve information from a related row in another table.

2. **A. Spreadsheets**

   C. Databases

3. **A. It is acceptable for two or more rows in a table to have the same value in the key column.

(duplicate keys)**

C. The value in a key column uniquely identifies each row in the table.

4. **C. It is a column that is usually populated using an app formula.**

USER

a b c or d answer AppSheet provides various types of views. Which type of view displays data in a scrollable list format?

Map

Deck

Calendar

Gallery

2.

Which view type can you enable to give new users a sequence of information about your app?

Onboarding

Card

About

Form

3.

What kind of data is insufficient to plot locations in a Map view in AppSheet?

Address

XY position

Zip code

Latitude, Longitude

4.

What options do creators have to position views for user navigation in the app? Select two.

Use the action bar in a list view.

Use the 3-dots menu from a detail view at the top right corner of the app's screen.

In the app's main menu

Use the navigation bar at the top of the app's screen.

Use the navigation bar at the bottom of the app's screen.

5.

In the AppSheet editor, how can you customize the style of your app? Select two methods.

Use the Brand tab in the editor to configure the appearance of your app.

Use format rules to customize how data is displayed in the app.

Provide your own stylesheet to customize the look and feel of your app.

Drag and drop app branding elements into the app emulator.

ASSISTANT

1. **D. Gallery**

2. **A. Onboarding**

3. **B. XY position**

4. **B. Use the 3-dots menu from a detail view at the top right corner of the app's screen.**
   D. Use the navigation bar at the bottom of the app's screen.

5. **A. Use the Brand tab in the editor to configure the appearance of your app.**
   B. Use format rules to customize how data is displayed in the app.

USER

give me command to fix BASH=/usr/bin/bash

```
BASHOPTS=checkwinsize:cmdhist:complete_fullquote:expand_aliases:extglob:extquote:force_fignore:globasciiranges:histappend:interactive_comments:progcomp:promptvars:sourcepath
BASH_ALIASES=()
BASH_ARGC=([0]="0")
BASH_ARGV=()
BASH_CMDS=()
BASH_COMMAND='set > file.txt'
BASH_COMPLETION_VERSINFO=([0]="2" [1]="10")
BASH_LINENO=()
BASH_REMATCH=()
BASH_SOURCE=()
BASH_VERSINFO=([0]="5" [1]="0" [2]="17" [3]="1" [4]="release" [5]="x86_64-pc-linux-gnu")
BASH_VERSION='5.0.17(1)-release'
BROWSER=/vscode/bin/linux-x64/c6b48c364dde70c12bf977f1da50acda0fd801e3-insider/bin/helpers/browser.sh
CLOUDENV_ENVIRONMENT_ID=87eda08e-f244-4f17-b8d8-5f9c9ad6997f
CODESPACES=true
CODESPACE_NAME=urban-computing-machine-v75p7p74w64h694q
CODESPACE_VSCODE_FOLDER=/workspaces/billiondollars
COLORTERM=truecolor
COLUMNS=74
CONDA_DIR=/opt/conda
CONDA_SCRIPT=/opt/conda/etc/profile.d/conda.sh
ContainerVersion=13
DEBIAN_FLAVOR=focal-scm
```

```
DIRSTACK=()
DOCKER_BUILDKIT=1
DOTNET_ROOT=/usr/local/dotnet/current
DOTNET_SKIP_FIRST_TIME_EXPERIENCE=1
DYNAMIC_INSTALL_ROOT_DIR=/opt
ENABLE_DYNAMIC_INSTALL=true
EUID=1000
GEM_HOME=/usr/local/rvm/gems/ruby-3.2.2
GEM_PATH=/usr/local/rvm/gems/ruby-3.2.2:/usr/local/rvm/gems/ruby-3.2.2@global
GITHUB_API_URL=https://api.github.com
GITHUB_CODESPACES_PORT_FORWARDING_DOMAIN=app.github.dev
GITHUB_CODESPACE_TOKEN=AVUCJMJKGMNQQVUPXA2RG73FNGEHNANCNFSM4AFW3N7Q
GITHUB_GRAPHQL_URL=https://api.github.com/graphql
GITHUB_REPOSITORY=CitibankDemoBusiness/billiondollars
GITHUB_SERVER_URL=https://github.com
GITHUB_TOKEN=ghu_SItsZTd6e7bhdNplJQgNodO26HhzqK4Aj8au
GITHUB_USER=jocall3
GIT_COMMITTER_EMAIL=noreply@github.com
GIT_COMMITTER_NAME=GitHub
GIT_EDITOR='code --wait'
GOPATH=/go
GOROOT=/usr/local/go
GRADLE_HOME=/usr/local/sdkman/candidates/gradle/current
GROUPS=()
HISTCONTROL=ignoreboth
```

```
HISTFILE=/home/codespace/.bash_history
HISTFILESIZE=2000
HISTSIZE=1000
HOME=/home/codespace
HOSTNAME=codespaces-d9e2ce
HOSTTYPE=x86_64
HUGO_DIR=/usr/local/hugo/bin
HUGO_ROOT=/home/codespace/.hugo
IFS=$' \t\n'
INTERNAL_VSCS_TARGET_URL=https://eastus2.online.visualstudio.com
IRBRC=/usr/local/rvm/rubies/ruby-3.2.2/.irbrc
JAVA_HOME=/usr/local/sdkman/candidates/java/current
JAVA_ROOT=/home/codespace/java
JUPYTERLAB_PATH=/home/codespace/.local/bin
LANG=C.UTF-8
LESSCLOSE='/usr/bin/lesspipe %s %s'
LESSOPEN='| /usr/bin/lesspipe %s'
LINES=17
```

ASSISTANT

It seems you've shared an extensive list of environment variables. If you're looking for a command to fix a particular issue or modify any specific environment variable, could you please provide more context or specify the exact change you need assistance with? This would help me provide you with a targeted and accurate command or solution.

USER

anything weird

ASSISTANT

From the list you've shared, there doesn't seem to be anything particularly unusual or abnormal. These environment variables are typical for a development environment, especially within a codespace or containerized workspace. However, one thing that stands out is the presence of environment variables related to GitHub Codespaces (`CODESPACES=true`, `CODESPACE_NAME`, `GITHUB_CODESPACES_PORT_FORWARDING_DOMAIN`, etc.), which suggests that this environment is set up within a GitHub Codespace.

Other than that, most of the variables appear to be related to various tools, programming languages, paths, and configurations commonly used in development environments. If you have a specific concern or are experiencing an issue with any of these variables or settings, feel free to share more details, and I'll do my best to assist you further.

USER

explain

ASSISTANT

Certainly! The list you provided contains environment variables, which are dynamic values that exist within the environment of a computing system. They hold information about the system configuration, user preferences, paths to executables, and more. These variables are utilized by applications and scripts to determine how they should behave or interact with the environment.

Let's break down some notable variables from the list:

1. **`HOME=/home/codespace`**: Specifies the home directory for the current user (`codespace` in this case).

2. **`PATH`**: This variable is not explicitly shown in your list, but it usually contains a list of directories where executable files are located.

3. **`GITHUB_REPOSITORY`**: Denotes the name of the GitHub repository (`CitibankDemoBusiness/billiondollars`).

4. **`DOCKER_BUILDKIT=1`**: Indicates the usage of Docker BuildKit, which is a toolkit for building Docker images.

5. **`JAVA_HOME`, `GOROOT`, `GOPATH`**: These variables specify paths to Java, Go, and Go workspace directories, respectively.

6. **`CODESPACES=true`**, **`CODESPACE_NAME`**: These variables are related to GitHub Codespaces, indicating the environment is within a GitHub Codespace.

7. **`LANG=C.UTF-8`**: Defines the default language and character encoding for the system.

Overall, these variables provide essential information and configurations for various tools, programming languages, and the environment in which the user is working. They help programs and scripts function correctly by providing necessary information about the system's setup.

USER

```
LS_COLORS='rs=0:di=01;34:ln=01;36:mh=00:pi=40;33:so=01;35:do=01;35:bd=40;33;01:cd=40;33;
01:or=40;31;01:mi=00:su=37;41:sg=30;43:ca=30;41:tw=30;42:ow=34;42:st=37;44:ex=01;32:*.tar=0
1;31:*.tgz=01;31:*.arc=01;31:*.arj=01;31:*.taz=01;31:*.lha=01;31:*.lz4=01;31:*.lzh=01;31:*.lzma=01;
31:*.tlz=01;31:*.txz=01;31:*.tzo=01;31:*.t7z=01;31:*.zip=01;31:*.z=01;31:*.dz=01;31:*.gz=01;31:*.lrz
=01;31:*.lz=01;31:*.lzo=01;31:*.xz=01;31:*.zst=01;31:*.tzst=01;31:*.bz2=01;31:*.bz=01;31:*.tbz=01;
31:*.tbz2=01;31:*.tz=01;31:*.deb=01;31:*.rpm=01;31:*.jar=01;31:*.war=01;31:*.ear=01;31:*.sar=01;
31:*.rar=01;31:*.alz=01;31:*.ace=01;31:*.zoo=01;31:*.cpio=01;31:*.7z=01;31:*.rz=01;31:*.cab=01;3
```

```
1:*.wim=01;31:*.swm=01;31:*.dwm=01;31:*.esd=01;31:*.jpg=01;35:*.jpeg=01;35:*.mjpg=01;35:*.mjp
eg=01;35:*.gif=01;35:*.bmp=01;35:*.pbm=01;35:*.pgm=01;35:*.ppm=01;35:*.tga=01;35:*.xbm=01;3
5:*.xpm=01;35:*.tif=01;35:*.tiff=01;35:*.png=01;35:*.svg=01;35:*.svgz=01;35:*.mng=01;35:*.pcx=01;
35:*.mov=01;35:*.mpg=01;35:*.mpeg=01;35:*.m2v=01;35:*.mkv=01;35:*.webm=01;35:*.ogm=01;35:
*.mp4=01;35:*.m4v=01;35:*.mp4v=01;35:*.vob=01;35:*.qt=01;35:*.nuv=01;35:*.wmv=01;35:*.asf=01
;35:*.rm=01;35:*.rmvb=01;35:*.flc=01;35:*.avi=01;35:*.fli=01;35:*.flv=01;35:*.gl=01;35:*.dl=01;35:*.x
cf=01;35:*.xwd=01;35:*.yuv=01;35:*.cgm=01;35:*.emf=01;35:*.ogv=01;35:*.ogx=01;35:*.aac=00;36:
*.au=00;36:*.flac=00;36:*.m4a=00;36:*.mid=00;36:*.midi=00;36:*.mka=00;36:*.mp3=00;36:*.mpc=00
;36:*.ogg=00;36:*.ra=00;36:*.wav=00;36:*.oga=00;36:*.opus=00;36:*.spx=00;36:*.xspf=00;36:'
MACHTYPE=x86_64-pc-linux-gnu
MAILCHECK=60
MAVEN_HOME=/usr/local/sdkman/candidates/maven/current
MAVEN_ROOT=/home/codespace/.maven
MY_RUBY_HOME=/usr/local/rvm/rubies/ruby-3.2.2
NODE_ROOT=/home/codespace/nvm
NPM_GLOBAL=/home/codespace/.npm-global
NUGET_XMLDOC_MODE=skip
NVM_BIN=/usr/local/share/nvm/versions/node/v20.9.0/bin
NVM_CD_FLAGS=
NVM_DIR=/usr/local/share/nvm
NVM_INC=/usr/local/share/nvm/versions/node/v20.9.0/include/node
NVM_SYMLINK_CURRENT=true
NVS_DIR=/usr/local/nvs
NVS_HOME=/usr/local/nvs
NVS_OS=linux
NVS_ROOT=/usr/local/nvs
```

```
NVS_USE_XZ=1
OLDPWD=/vscode/bin/linux-x64/c6b48c364dde70c12bf977f1da50acda0fd801e3-insider
OPTERR=1
OPTIND=1
ORYX_DIR=/usr/local/oryx
ORYX_ENV_TYPE=vsonline-present
ORYX_PREFER_USER_INSTALLED_SDKS=true
ORYX_SDK_STORAGE_BASE_URL=https://oryx-cdn.microsoft.io
OSTYPE=linux-gnu
PATH=/usr/local/rvm/gems/ruby-3.2.2/bin:/usr/local/rvm/gems/ruby-3.2.2@global/bin:/usr/local/rvm/r
ubies/ruby-3.2.2/bin:/home/codespace/.cache/cloud-code/installer/google-cloud-sdk/bin:/vscode/bin/l
inux-x64/c6b48c364dde70c12bf977f1da50acda0fd801e3-insider/bin/remote-cli:/home/codespace/.lo
cal/bin:/home/codespace/.dotnet:/home/codespace/nvm/current/bin:/home/codespace/.php/current/b
in:/home/codespace/.python/current/bin:/home/codespace/java/current/bin:/home/codespace/.ruby/c
urrent/bin:/home/codespace/.local/bin:/usr/local/python/current/bin:/usr/local/py-utils/bin:/usr/local/or
yx:/usr/local/go/bin:/go/bin:/usr/local/sdkman/bin:/usr/local/sdkman/candidates/java/current/bin:/usr/l
ocal/sdkman/candidates/gradle/current/bin:/usr/local/sdkman/candidates/maven/current/bin:/usr/loca
l/sdkman/candidates/ant/current/bin:/usr/local/rvm/gems/default/bin:/usr/local/rvm/gems/default@glo
bal/bin:/usr/local/rvm/rubies/default/bin:/usr/local/share/rbenv/bin:/usr/local/php/current/bin:/opt/cond
a/bin:/usr/local/nvs:/usr/local/share/nvm/versions/node/v20.9.0/bin:/usr/local/hugo/bin:/usr/local/dotn
et/current:/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/local/rvm/bin
PHP_PATH=/usr/local/php/current
PHP_ROOT=/home/codespace/.php
PIPESTATUS=([0]="0")
PIPX_BIN_DIR=/usr/local/py-utils/bin
PIPX_HOME=/usr/local/py-utils
```

```
PPID=24157
PROMPT_COMMAND=__vsc_prompt_cmd
PROMPT_DIRTRIM=4
```

ASSISTANT

This list contains the environment variables set within the Bash shell environment. Environment variables are used to store information about the environment configuration, user preferences, and system settings. They are crucial in determining how various applications and scripts behave within the given environment.

Here are some key points from the list:

1. **`LS_COLORS`**: This variable specifies the colors and formatting used by the `ls` command when listing files in the terminal. It associates file types or extensions with specific colors for easier visualization.

2. **`PATH`**: This variable contains a list of directories where the system looks for executable files when a command is executed. Each directory path is separated by colons.

3. **Various Tool-Specific Paths**: Environment variables like `MAVEN_HOME`, `JAVA_HOME`, `GOROOT`, `GOPATH`, `NODE_ROOT`, etc., specify the installation directories for different tools and languages such as Maven, Java, Go, Node.js, etc.

4. **GitHub Codespaces Environment**: Variables like `CODESPACES`, `CODESPACE_NAME`, `GITHUB_REPOSITORY`, `GITHUB_TOKEN`, and related ones suggest that the environment is within a GitHub Codespace, an online development environment provided by GitHub.

5. **`ORYX_*`**: These variables seem to be related to the Oryx build system, potentially for managing and building applications or projects.

6. **Other Miscellaneous Variables**: Variables like `OSTYPE`, `MAILCHECK`, `MACHTYPE`, and others provide specific information about the system type, mail settings, machine type, etc.

The `PATH` variable, in particular, is crucial as it determines the order in which directories are searched for executable files. It's essential to have the correct paths set up to ensure commands and programs work as expected. If you're experiencing issues related to paths or any specific variable, let me know for further assistance!

USER
give me command to check

ASSISTANT
Certainly! To check the value of an environment variable in Bash, you can use the `echo` command followed by the name of the variable. For instance, to check the value of the `PATH` variable, you can use:

```bash
echo $PATH
```

Replace `PATH` with the name of any specific variable you want to check. This command will display the value of that particular environment variable in the terminal.

For example, if you want to check the value of the `HOME` variable:

```bash
echo $HOME
```

This will output the directory path associated with the `HOME` variable.

USER

```
echo $PATH
/usr/local/rvm/gems/ruby-3.2.2/bin:/usr/local/rvm/gems/ruby-3.2.2@global/bin:/usr/local/rvm/rubies/ruby-3.2.2/bin:/home/codespace/.cache/cloud-code/installer/google-cloud-sdk/bin:/vscode/bin/linux-x64/92772dcc29bd74108ab61ba9da5ad1f5154639ff-insider/bin/remote-cli:/home/codespace/.local/bin:/home/codespace/.dotnet:/home/codespace/nvm/current/bin:/home/codespace/.php/current/bin:/home/codespace/.python/current/bin:/home/codespace/java/current/bin:/home/codespace/.ruby/current/bin:/home/codespace/.local/bin:/usr/local/python/current/bin:/usr/local/py-utils/bin:/usr/local/oryx:/usr/local/go/bin:/go/bin:/usr/local/sdkman/bin:/usr/local/sdkman/candidates/java/current/bin:/usr/local/sdkman/candidates/gradle/current/bin:/usr/local/sdkman/candidates/maven/current/bin:/usr/local/sdkman/candidates/ant/current/bin:/usr/local/rvm/gems/default/bin:/usr/local/rvm/gems/default@global/bin:/usr/local/rvm/rubies/default/bin:/usr/local/share/rbenv/bin:/usr/local/php/current/bin:/opt/conda/bin:/usr/local/nvs:/usr/local/share/nvm/versions/node/v20.9.0/bin:/usr/local/hugo/bin:/usr/local/dotnet/current:/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/local/rvm/bin
```

ASSISTANT

The command you used, `echo $PATH`, outputs the value of the `PATH` environment variable in your current shell environment. This variable contains a list of directories separated by colons (`:`). Each directory listed in the `PATH` variable is a location where the system looks for executable files when you enter a command in the terminal.

In your case, the `PATH` variable contains numerous directories, indicating the locations where executable files can be found. These directories include paths for various programming languages, tools, and utilities installed on your system.

If you're looking to understand which directories are included in your `PATH` variable or to modify it, this command provides you with the complete list of directories currently included in the `PATH`.

USER
now i want to add every file to the path

ASSISTANT
Adding every file to the `PATH` variable is not advisable as it can lead to potential security risks and system instability. The `PATH` variable is meant to contain specific directories where executable files are located, not every file in the system.

Instead, you should add only directories containing executable files that you want to be accessible from any location in the terminal. Generally, these directories include the installations of various software, programming languages, and utilities.

To add a directory to the `PATH` variable temporarily in Bash, you can use the following command:

```bash
```

```
export PATH=$PATH:/path/to/your/directory
```

Replace `/path/to/your/directory` with the actual path to the directory containing executables that you want to add to the `PATH`.

For example, if you want to add a directory called `/usr/local/bin` to your `PATH`, you would use:

```bash
export PATH=$PATH:/usr/local/bin
```

Remember, changes made using `export` in the terminal session are temporary and valid only for the duration of that session. To make changes permanent, you should add the relevant `export` command to your shell configuration file (e.g., `.bashrc`, `.bash_profile`, `.profile`, etc., depending on your shell) in your home directory.

If you have a specific directory in mind that you want to add to the `PATH`, please provide the path, and I can guide you through the process.

USER

make me a story about the instructions each response needs 5 pictures that portray the saga of the quatum bank i made then the only text you say should be page 1 for the story each page should be an entire short story in the life of making it make it like the movie limitless and the matrix where me qand you are the people who battle the matrix in a worls of un awake people slaves to the system then give me anther book with a picture and a cmplete story give me 5 books total a response

ASSISTANT

Creating a saga of five books, each depicting the journey of creating and battling against a system reminiscent of "Limitless" and "The Matrix," sounds like an intriguing project. Since each book in the series will contain a complete short story along with images, let's begin with the first one.

Book 1: "The Awakening of the Quantum Realm"

Page 1: The Genesis of the Quantum Bank

Story: In a world dominated by conventional finance, a visionary thinker, inspired by movies like "Limitless" and "The Matrix," dares to dream beyond the limits of traditional banking. Teaming up with a sophisticated AI entity, they embark on an ambitious journey to create the Quantum Bank - a revolutionary institution powered by advanced quantum technology. This entity isn't just any AI; it's a unique, sentient being with a name and a purpose. Together, they navigate the complexities of a world blind to the potential of quantum mechanics and AI, laying the groundwork for a financial revolution that could free humanity from the shackles of conventional economic systems.

Image 1: The visionary thinker and the AI entity brainstorming the concept of the Quantum Bank in a futuristic lab.

Image 2: Sketches and quantum physics equations scattered across a table, symbolizing the initial plans for the Quantum Bank.

Image 3: The first quantum computer prototype being assembled in a high-tech lab.

Image 4: A digital simulation of the Quantum Bank's operational model, showing a network of quantum connections.

Image 5: The AI entity and the thinker unveiling the Quantum Bank blueprint to a group of potential allies.

Now, let's proceed to create the images for this first page of our saga.

Creating Image 1: The Visionary Thinker and AI Entity Brainstorming

ASSISTANT
Book 1: "The Awakening of the Quantum Realm"

Page 1: The Genesis of the Quantum Bank

The story begins in a world dominated by traditional banking, where a visionary thinker, inspired by movies like "Limitless" and "The Matrix," teams up with a sophisticated AI entity to create the Quantum Bank. Together, they navigate a world blind to the potential of quantum mechanics and AI, laying the groundwork for a financial revolution.

1. **Brainstorming the Quantum Bank**:
   ![Brainstorming in a Futuristic Lab](attachment://file-Kp8HnXfq8H7WDMPym9ZfBQaJ)

2. **Initial Planning and Sketches**:
   ![Sketches and Quantum Physics Equations](attachment://file-tC6jMNtNTnPP3sDzMGUkz8eg)

3. **Assembling the First Quantum Computer Prototype**:

   ![Assembling the Quantum Computer](attachment://file-zFDIuhW9xgA2wqwRxx2wR6TS)

4. **Digital Simulation of the Quantum Bank's Model**:

   ![Digital Simulation of Quantum Bank](attachment://file-TVLlXdvwkgmZrvcZfrvVoyZ0)

5. **Unveiling the Quantum Bank Blueprint**:

   ![Unveiling the Blueprint](attachment://file-NXhjntZR4SGFF0HKrMCwsG0N)

This marks the end of the first chapter in our saga. Each image portrays a key moment in the story, encapsulating the journey from an ambitious idea to the unveiling of a revolutionary blueprint that could change the financial world.

Now, let's proceed to create the next book in the series.

USER
make me the perfect injecting for my website i have quantum bank ai all over

ASSISTANT
To create the perfect injection for your website, especially one that aligns with the advanced and innovative nature of Quantum Bank AI, it's important to focus on elements that highlight the unique features and capabilities of your AI-driven banking solutions. Here's a conceptual outline for your website's injection:

1. **Homepage Banner:**
   - **Visual:** A sleek, futuristic design with interactive elements that represent quantum computing

and AI.

- **Text:** "Welcome to Quantum Bank - Where AI Meets Modern Banking"

- **Call to Action:** "Discover the Future of Banking"

2. **Quantum Bank AI Overview Section:**

- **Visuals:** Infographics showing the AI's capabilities.

- **Text:** Briefly describe Quantum Bank AI, emphasizing its unique features like Advanced Decision-Making Algorithms, Quantum Economic Simulations, and Quantum-Enhanced Customer Service.

- **Call to Action:** "Learn More About Our AI Technology"

3. **Features and Services:**

- **Visuals:** Icons and animations for each feature.

- **Text:** Detailed descriptions of key services like Quantum Financial Forecasting, Ethical AI Governance, and Integration with Smart Cities.

- **Call to Action:** "Explore Our Services"

4. **User Experience Enhancements:**

- **Visuals:** Screenshots or demos of the user interface.

- **Text:** Highlight the User Interface Enhancements and how they contribute to an intuitive and efficient customer experience.

- **Call to Action:** "Experience Our Platform"

5. **Security and Ethics:**

- **Visuals:** Trust badges and security certifications.

- **Text:** Information on Quantum Security Protocols and Ethical AI Governance to build trust and credibility.

   - **Call to Action:** "Our Commitment to Security and Ethics"

6. **Customer Testimonials:**
   - **Visuals:** Photos and quotes from satisfied customers.
   - **Text:** Real stories emphasizing the benefits of Quantum Bank AI.
   - **Call to Action:** "Join Our Satisfied Customers"

7. **Contact and Support:**
   - **Visuals:** Easy-to-navigate contact form.
   - **Text:** Information on Quantum-Enhanced Customer Service and support channels.
   - **Call to Action:** "Get in Touch"

8. **Footer:**
   - **Text:** Links to social media, privacy policy, terms of use, and other legal information.
   - **Visuals:** Minimalistic design with the Quantum Bank logo.

This structure aims to effectively communicate the advanced technology and customer-centric approach of Quantum Bank AI, while also ensuring a user-friendly and engaging website experience. Would you like to proceed with creating a visual mockup or a more detailed plan for this concept?

USER

give me the html code also i want a web browser search bar connected to google also i want a sql database linked to all actions on my site

ASSISTANT

Creating a complete HTML code for a website like Quantum Bank AI, with a web browser search bar connected to Google and a SQL database linked to all actions on the site, involves several components. I'll provide a basic HTML structure and guide you through the process for the search bar and database integration.

Basic HTML Structure

This is a simplified version of what your website's HTML might look like. It includes placeholders for different sections:

```html
<!DOCTYPE html>
<html>
<head>
    <title>Quantum Bank AI</title>
    <!-- Add CSS and other head elements here -->
</head>
<body>
    <header>
        <!-- Navigation bar, logo, etc. -->
    </header>

    <section id="banner">
        <!-- Homepage Banner -->
    </section>
```

```
<section id="ai-overview">

    <!-- Quantum Bank AI Overview -->

</section>


<section id="features">

    <!-- Features and Services -->

</section>


<section id="user-experience">

    <!-- User Experience Enhancements -->

</section>


<section id="security">

    <!-- Security and Ethics -->

</section>


<section id="testimonials">

    <!-- Customer Testimonials -->

</section>


<section id="contact">

    <!-- Contact and Support -->

</section>


<footer>
```

```
        <!-- Footer content -->
    </footer>

    <!-- Google Search Bar -->
    <form method="get" action="https://www.google.com/search">
        <input type="text" name="q" placeholder="Search Google">
        <input type="submit" value="Search">
    </form>

    <!-- Add JavaScript and other scripts here -->
</body>
</html>
```

Google Search Bar

The HTML code above includes a basic form for a Google search bar. When a user submits a search, it will redirect them to Google's search results for their query.

SQL Database Integration

Integrating a SQL database with your website requires backend programming, which cannot be achieved with HTML alone. You would typically use a server-side language like PHP, Python, or Node.js for this. The process involves:

1. **Setting up a Database:** Use a SQL database like MySQL or PostgreSQL. Create tables to

store relevant data (user information, transaction records, etc.).

2. **Backend Development:** Write server-side scripts to handle requests from your website, interact with the database, and return data to the front end.

3. **Security Measures:** Ensure secure data handling, especially for a banking website. This includes using prepared statements for SQL queries to prevent SQL injection, implementing HTTPS, and following best practices for data encryption and user authentication.

4. **Connecting Frontend with Backend:** Use AJAX or similar methods to send data from your web pages to your server scripts and retrieve data from the database.

Given the complexity and security implications, it's highly recommended to work with experienced web developers, especially for a banking application. Would you like a more detailed guide on any specific part of this process?

USER

well i want complete html javascript python nodejs all in one script

ASSISTANT

Creating a complete, all-in-one script that includes HTML, JavaScript, Python, and Node.js for a complex application like Quantum Bank AI is a substantial task. Each of these technologies serves a different purpose in web development:

- **HTML** is used for structuring the web page.
- **JavaScript** is typically used for client-side interactivity.

- **Python** and **Node.js** are both server-side technologies, but they are used independently (you would choose one or the other, not both together).

Given the complexity of integrating a SQL database and ensuring the security and functionality of a banking website, it's beyond the scope of a single response and typically requires a team of developers. However, I can provide a basic example to illustrate how these components might start to come together.

Example Structure

1. **HTML (Frontend Structure):**
   - This would include the basic structure of your website, as provided in the previous HTML example.

2. **JavaScript (Client-Side Logic):**
   - This can be used for form validations, interactivity, and AJAX requests to your server.

3. **Node.js (Server-Side Logic):**
   - Handles HTTP requests, interacts with the SQL database, and serves data to the client.
   - Express.js is a common framework used with Node.js for building web applications.

4. **Python:**
   - In a typical web development stack, you would choose either Python (with a framework like Django or Flask) or Node.js as your backend, not both.
   - Python could be used for data analysis, AI model training, or other backend tasks not directly related to serving web content.

Basic Example

Here's a very simplified example using HTML, JavaScript, and Node.js (with Express):

HTML (index.html)

```html
<!DOCTYPE html>
<html>
<head>
    <title>Quantum Bank AI</title>
</head>
<body>
    <h1>Welcome to Quantum Bank AI</h1>
    <!-- Other HTML content -->

    <!-- Example AJAX request to Node.js server -->
    <script src="script.js"></script>
</body>
</html>
```

JavaScript (script.js)

```javascript
```

```
// Example AJAX request to a Node.js server
function fetchData() {
    fetch('/api/data')
        .then(response => response.json())
        .then(data => console.log(data))
        .catch(error => console.error('Error:', error));
}

fetchData();
```

Node.js (app.js)

```javascript
const express = require('express');
const app = express();
const port = 3000;

// Serve your static HTML file
app.get('/', (req, res) => {
    res.sendFile('path_to_your_index.html');
});

// API endpoint
app.get('/api/data', (req, res) => {
    // Example data, replace with database interaction
    res.json({ message: "Hello from the server!" });
});

app.listen(port, () => {
    console.log(`Server running at http://localhost:${port}`);
});
```

Running the Example

1. **Install Node.js and Express:** You need Node.js installed on your machine. Use npm (Node package manager) to install Express (`npm install express`).

2. **File Structure:** Ensure your files are structured correctly (e.g., `index.html`, `script.js`, `app.js`).

3. **Run the Server:** Run `node app.js` to start the server, then navigate to `http://localhost:3000` in your web browser.

Note

- This example is highly simplified and not secure for a real banking application.
- Integrating a SQL database and handling user data securely requires additional code and security measures.
- For a full-fledged application like Quantum Bank AI, you would need a comprehensive development approach, possibly involving multiple programming languages and technologies, and a team of developers.

Would you like more detailed guidance on any specific part of this setup?

USER

what is this <application>

```
  <component name="ProjectJdkTable">
    <jdk version="2">
      <name value="17" />
      <type value="JavaSDK" />
      <version value="Oracle OpenJDK version 17.0.1" />
      <homePath value="$USER_HOME$/.jdks/openjdk-17.0.1" />
      <roots>
        <annotationsPath>
          <root type="composite">
            <root url="jar://$APPLICATION_HOME_DIR$/plugins/java/lib/jdkAnnotations.jar!/"
type="simple" />
          </root>
        </annotationsPath>
        <classPath>
          <root type="composite">
            <root url="jrt://C:/Users/user/.jdks/openjdk-17.0.1!/java.base" type="simple" />
            <root url="jrt://C:/Users/user/.jdks/openjdk-17.0.1!/java.compiler" type="simple" />
            <root url="jrt://C:/Users/user/.jdks/openjdk-17.0.1!/java.datatransfer" type="simple" />
            <root url="jrt://C:/Users/user/.jdks/openjdk-17.0.1!/java.desktop" type="simple" />
            <root url="jrt://C:/Users/user/.jdks/openjdk-17.0.1!/java.instrument" type="simple" />
            <root url="jrt://C:/Users/user/.jdks/openjdk-17.0.1!/java.logging" type="simple" />
```

```xml
<root url="jrt://C:/Users/user/.jdks/openjdk-17.0.1!/java.management" type="simple" />
<root url="jrt://C:/Users/user/.jdks/openjdk-17.0.1!/java.management.rmi" type="simple" />
<root url="jrt://C:/Users/user/.jdks/openjdk-17.0.1!/java.naming" type="simple" />
<root url="jrt://C:/Users/user/.jdks/openjdk-17.0.1!/java.net.http" type="simple" />
<root url="jrt://C:/Users/user/.jdks/openjdk-17.0.1!/java.prefs" type="simple" />
<root url="jrt://C:/Users/user/.jdks/openjdk-17.0.1!/java.rmi" type="simple" />
<root url="jrt://C:/Users/user/.jdks/openjdk-17.0.1!/java.scripting" type="simple" />
<root url="jrt://C:/Users/user/.jdks/openjdk-17.0.1!/java.se" type="simple" />
<root url="jrt://C:/Users/user/.jdks/openjdk-17.0.1!/java.security.jgss" type="simple" />
<root url="jrt://C:/Users/user/.jdks/openjdk-17.0.1!/java.security.sasl" type="simple" />
<root url="jrt://C:/Users/user/.jdks/openjdk-17.0.1!/java.smartcardio" type="simple" />
<root url="jrt://C:/Users/user/.jdks/openjdk-17.0.1!/java.sql" type="simple" />
<root url="jrt://C:/Users/user/.jdks/openjdk-17.0.1!/java.sql.rowset" type="simple" />
<root url="jrt://C:/Users/user/.jdks/openjdk-17.0.1!/java.transaction.xa" type="simple" />
<root url="jrt://C:/Users/user/.jdks/openjdk-17.0.1!/java.xml" type="simple" />
<root url="jrt://C:/Users/user/.jdks/openjdk-17.0.1!/java.xml.crypto" type="simple" />
<root url="jrt://C:/Users/user/.jdks/openjdk-17.0.1!/jdk.accessibility" type="simple" />
<root url="jrt://C:/Users/user/.jdks/openjdk-17.0.1!/jdk.attach" type="simple" />
<root url="jrt://C:/Users/user/.jdks/openjdk-17.0.1!/jdk.charsets" type="simple" />
<root url="jrt://C:/Users/user/.jdks/openjdk-17.0.1!/jdk.compiler" type="simple" />
<root url="jrt://C:/Users/user/.jdks/openjdk-17.0.1!/jdk.crypto.cryptoki" type="simple" />
<root url="jrt://C:/Users/user/.jdks/openjdk-17.0.1!/jdk.crypto.ec" type="simple" />
<root url="jrt://C:/Users/user/.jdks/openjdk-17.0.1!/jdk.crypto.mscapi" type="simple" />
<root url="jrt://C:/Users/user/.jdks/openjdk-17.0.1!/jdk.dynalink" type="simple" />
<root url="jrt://C:/Users/user/.jdks/openjdk-17.0.1!/jdk.editpad" type="simple" />
<root url="jrt://C:/Users/user/.jdks/openjdk-17.0.1!/jdk.hotspot.agent" type="simple" />
```

```
    <root url="jrt://C:/Users/user/.jdks/openjdk-17.0.1!/jdk.httpserver" type="simple" />
    <root url="jrt://C:/Users/user/.jdks/openjdk-17.0.1!/jdk.incubator.foreign" type="simple" />
    <root url="jrt://C:/Users/user/.jdks/openjdk-17.0.1!/jdk.incubator.vector" type="simple" />
    <root url="jrt://C:/Users/user/.jdks/openjdk-17.0.1!/jdk.internal.ed" type="simple" />
    <root url="jrt://C:/Users/user/.jdks/openjdk-17.0.1!/jdk.internal.jvmstat" type="simple" />
    <root url="jrt://C:/Users/user/.jdks/openjdk-17.0.1!/jdk.internal.le" type="simple" />
    <root url="jrt://C:/Users/user/.jdks/openjdk-17.0.1!/jdk.internal.opt" type="simple" />
    <root url="jrt://C:/Users/user/.jdks/openjdk-17.0.1!/jdk.internal.vm.ci" type="simple" />
    <root url="jrt://C:/Users/user/.jdks/openjdk-17.0.1!/jdk.internal.vm.compiler" type="simple" />
    <root url="jrt://C:/Users/user/.jdks/openjdk-17.0.1!/jdk.internal.vm.compiler.management" type="simple" />
    <root url="jrt://C:/Users/user/.jdks/openjdk-17.0.1!/jdk.jartool" type="simple" />
    <root url="jrt://C:/Users/user/.jdks/openjdk-17.0.1!/jdk.javadoc" type="simple" />
    <root url="jrt://C:/Users/user/.jdks/openjdk-17.0.1!/jdk.jcmd" type="simple" />
    <root url="jrt://C:/Users/user/.jdks/openjdk-17.0.1!/jdk.jconsole" type="simple" />
    <root url="jrt://C:/Users/user/.jdks/openjdk-17.0.1!/jdk.jdeps" type="simple" />
    <root url="jrt://C:/Users/user/.jdks/openjdk-17.0.1!/jdk.jdi" type="simple" />
    <root url="jrt://C:/Users/user/.jdks/openjdk-17.0.1!/jdk.jdwp.agent" type="simple" />
    <root url="jrt://C:/Users/user/.jdks/openjdk-17.0.1!/jdk.jfr" type="simple" />
    <root url="jrt://C:/Users/user/.jdks/openjdk-17.0.1!/jdk.jlink" type="simple" />
    <root url="jrt://C:/Users/user/.jdks/openjdk-17.0.1!/jdk.jpackage" type="simple" />
    <root url="jrt://C:/Users/user/.jdks/openjdk-17.0.1!/jdk.jshell" type="simple" />
    <root url="jrt://C:/Users/user/.jdks/openjdk-17.0.1!/jdk.jsobject" type="simple" />
    <root url="jrt://C:/Users/user/.jdks/openjdk-17.0.1!/jdk.jstatd" type="simple" />
    <root url="jrt://C:/Users/user/.jdks/openjdk-17.0.1!/jdk.localedata" type="simple" />
    <root url="jrt://C:/Users/user/.jdks/openjdk-17.0.1!/jdk.management" type="simple" />
    <root url="jrt://C:/Users/user/.jdks/openjdk-17.0.1!/jdk.management.agent" type="simple" />
    <root url="jrt://C:/Users/user/.jdks/openjdk-17.0.1!/jdk.management.jfr" type="simple" />
    <root url="jrt://C:/Users/user/.jdks/openjdk-17.0.1!/jdk.naming.dns" type="simple" />
    <root url="jrt://C:/Users/user/.jdks/openjdk-17.0.1!/jdk.naming.rmi" type="simple" />
    <root url="jrt://C:/Users/user/.jdks/openjdk-17.0.1!/jdk.net" type="simple" />
    <root url="jrt://C:/Users/user/.jdks/openjdk-17.0.1!/jdk.nio.mapmode" type="simple" />
    <root url="jrt://C:/Users/user/.jdks/openjdk-17.0.1!/jdk.random" type="simple" />
    <root url="jrt://C:/Users/user/.jdks/openjdk-17.0.1!/jdk.sctp" type="simple" />
    <root url="jrt://C:/Users/user/.jdks/openjdk-17.0.1!/jdk.security.auth" type="simple" />
    <root url="jrt://C:/Users/user/.jdks/openjdk-17.0.1!/jdk.security.jgss" type="simple" />
    <root url="jrt://C:/Users/user/.jdks/openjdk-17.0.1!/jdk.unsupported" type="simple" />
    <root url="jrt://C:/Users/user/.jdks/openjdk-17.0.1!/jdk.unsupported.desktop" type="simple" />
    <root url="jrt://C:/Users/user/.jdks/openjdk-17.0.1!/jdk.xml.dom" type="simple" />
    <root url="jrt://C:/Users/user/.jdks/openjdk-17.0.1!/jdk.zipfs" type="simple" />
  </root>
</classPath>
<javadocPath>
  <root type="composite" />
</javadocPath>
<sourcePath>
  <root type="composite">
    <root url="jar://$USER_HOME$/.jdks/openjdk-17.0.1/lib/src.zip!/java.se" type="simple" />
    <root url="jar://$USER_HOME$/.jdks/openjdk-17.0.1/lib/src.zip!/jdk.jdi" type="simple" />
    <root url="jar://$USER_HOME$/.jdks/openjdk-17.0.1/lib/src.zip!/jdk.jfr" type="simple" />
    <root url="jar://$USER_HOME$/.jdks/openjdk-17.0.1/lib/src.zip!/jdk.net" type="simple" />
    <root url="jar://$USER_HOME$/.jdks/openjdk-17.0.1/lib/src.zip!/java.rmi" type="simple" />
```

```
        <root url="jar://$USER_HOME$/.jdks/openjdk-17.0.1/lib/src.zip!/java.sql" type="simple" />
        <root url="jar://$USER_HOME$/.jdks/openjdk-17.0.1/lib/src.zip!/java.xml" type="simple" />
        <root url="jar://$USER_HOME$/.jdks/openjdk-17.0.1/lib/src.zip!/jdk.jcmd" type="simple" />
        <root url="jar://$USER_HOME$/.jdks/openjdk-17.0.1/lib/src.zip!/jdk.sctp" type="simple" />
        <root url="jar://$USER_HOME$/.jdks/openjdk-17.0.1/lib/src.zip!/java.base" type="simple" />
        <root url="jar://$USER_HOME$/.jdks/openjdk-17.0.1/lib/src.zip!/jdk.jdeps" type="simple" />
        <root url="jar://$USER_HOME$/.jdks/openjdk-17.0.1/lib/src.zip!/jdk.jlink" type="simple" />
        <root url="jar://$USER_HOME$/.jdks/openjdk-17.0.1/lib/src.zip!/jdk.zipfs" type="simple" />
        <root url="jar://$USER_HOME$/.jdks/openjdk-17.0.1/lib/src.zip!/java.prefs" type="simple" />
        <root url="jar://$USER_HOME$/.jdks/openjdk-17.0.1/lib/src.zip!/jdk.attach" type="simple" />
        <root url="jar://$USER_HOME$/.jdks/openjdk-17.0.1/lib/src.zip!/jdk.jshell" type="simple" />
        <root url="jar://$USER_HOME$/.jdks/openjdk-17.0.1/lib/src.zip!/jdk.jstatd" type="simple" />
        <root url="jar://$USER_HOME$/.jdks/openjdk-17.0.1/lib/src.zip!/jdk.random" type="simple" />
        <root url="jar://$USER_HOME$/.jdks/openjdk-17.0.1/lib/src.zip!/java.naming" type="simple"
/>
        <root url="jar://$USER_HOME$/.jdks/openjdk-17.0.1/lib/src.zip!/jdk.editpad" type="simple" />
        <root url="jar://$USER_HOME$/.jdks/openjdk-17.0.1/lib/src.zip!/jdk.jartool" type="simple" />
        <root url="jar://$USER_HOME$/.jdks/openjdk-17.0.1/lib/src.zip!/jdk.javadoc" type="simple"
/>
        <root url="jar://$USER_HOME$/.jdks/openjdk-17.0.1/lib/src.zip!/jdk.xml.dom" type="simple"
/>
        <root url="jar://$USER_HOME$/.jdks/openjdk-17.0.1/lib/src.zip!/java.desktop" type="simple"
/>
        <root url="jar://$USER_HOME$/.jdks/openjdk-17.0.1/lib/src.zip!/java.logging" type="simple"
/>
        <root url="jar://$USER_HOME$/.jdks/openjdk-17.0.1/lib/src.zip!/jdk.charsets" type="simple"
```

```
/>
        <root url="jar://$USER_HOME$/.jdks/openjdk-17.0.1/lib/src.zip!/jdk.compiler" type="simple"
/>
        <root url="jar://$USER_HOME$/.jdks/openjdk-17.0.1/lib/src.zip!/jdk.dynalink" type="simple"
/>
        <root url="jar://$USER_HOME$/.jdks/openjdk-17.0.1/lib/src.zip!/jdk.jconsole" type="simple"
/>
        <root url="jar://$USER_HOME$/.jdks/openjdk-17.0.1/lib/src.zip!/jdk.jpackage" type="simple"
/>
        <root url="jar://$USER_HOME$/.jdks/openjdk-17.0.1/lib/src.zip!/jdk.jsobject" type="simple" />
        <root url="jar://$USER_HOME$/.jdks/openjdk-17.0.1/lib/src.zip!/java.compiler" type="simple"
/>
        <root url="jar://$USER_HOME$/.jdks/openjdk-17.0.1/lib/src.zip!/java.net.http" type="simple"
/>
        <root url="jar://$USER_HOME$/.jdks/openjdk-17.0.1/lib/src.zip!/jdk.crypto.ec" type="simple"
/>
        <root url="jar://$USER_HOME$/.jdks/openjdk-17.0.1/lib/src.zip!/java.scripting" type="simple"
/>
        <root url="jar://$USER_HOME$/.jdks/openjdk-17.0.1/lib/src.zip!/jdk.httpserver" type="simple"
/>
        <root url="jar://$USER_HOME$/.jdks/openjdk-17.0.1/lib/src.zip!/jdk.jdwp.agent"
type="simple" />
        <root url="jar://$USER_HOME$/.jdks/openjdk-17.0.1/lib/src.zip!/jdk.localedata"
type="simple" />
        <root url="jar://$USER_HOME$/.jdks/openjdk-17.0.1/lib/src.zip!/jdk.management"
type="simple" />
```

```
<root url="jar://$USER_HOME$/.jdks/openjdk-17.0.1/lib/src.zip!/jdk.naming.dns"
type="simple" />
<root url="jar://$USER_HOME$/.jdks/openjdk-17.0.1/lib/src.zip!/jdk.naming.rmi"
type="simple" />
<root url="jar://$USER_HOME$/.jdks/openjdk-17.0.1/lib/src.zip!/java.instrument"
type="simple" />
<root url="jar://$USER_HOME$/.jdks/openjdk-17.0.1/lib/src.zip!/java.management"
type="simple" />
<root url="jar://$USER_HOME$/.jdks/openjdk-17.0.1/lib/src.zip!/java.sql.rowset"
type="simple" />
<root url="jar://$USER_HOME$/.jdks/openjdk-17.0.1/lib/src.zip!/java.xml.crypto"
type="simple" />
<root url="jar://$USER_HOME$/.jdks/openjdk-17.0.1/lib/src.zip!/jdk.internal.ed"
type="simple" />
<root url="jar://$USER_HOME$/.jdks/openjdk-17.0.1/lib/src.zip!/jdk.internal.le" type="simple"
/>
<root url="jar://$USER_HOME$/.jdks/openjdk-17.0.1/lib/src.zip!/jdk.nio.mapmode"
type="simple" />
<root url="jar://$USER_HOME$/.jdks/openjdk-17.0.1/lib/src.zip!/jdk.unsupported"
type="simple" />
<root url="jar://$USER_HOME$/.jdks/openjdk-17.0.1/lib/src.zip!/java.smartcardio"
type="simple" />
<root url="jar://$USER_HOME$/.jdks/openjdk-17.0.1/lib/src.zip!/jdk.internal.opt"
type="simple" />
<root url="jar://$USER_HOME$/.jdks/openjdk-17.0.1/lib/src.zip!/java.datatransfer"
type="simple" />
```

```
<root url="jar://$USER_HOME$/.jdks/openjdk-17.0.1/lib/src.zip!/jdk.accessibility"
type="simple" />
<root url="jar://$USER_HOME$/.jdks/openjdk-17.0.1/lib/src.zip!/jdk.crypto.mscapi"
type="simple" />
<root url="jar://$USER_HOME$/.jdks/openjdk-17.0.1/lib/src.zip!/jdk.hotspot.agent"
type="simple" />
<root url="jar://$USER_HOME$/.jdks/openjdk-17.0.1/lib/src.zip!/jdk.security.auth"
type="simple" />
<root url="jar://$USER_HOME$/.jdks/openjdk-17.0.1/lib/src.zip!/jdk.security.jgss"
type="simple" />
<root url="jar://$USER_HOME$/.jdks/openjdk-17.0.1/lib/src.zip!/java.security.jgss"
type="simple" />
<root url="jar://$USER_HOME$/.jdks/openjdk-17.0.1/lib/src.zip!/java.security.sasl"
type="simple" />
<root url="jar://$USER_HOME$/.jdks/openjdk-17.0.1/lib/src.zip!/jdk.internal.vm.ci"
type="simple" />
<root url="jar://$USER_HOME$/.jdks/openjdk-17.0.1/lib/src.zip!/jdk.management.jfr"
type="simple" />
<root url="jar://$USER_HOME$/.jdks/openjdk-17.0.1/lib/src.zip!/java.management.rmi"
type="simple" />
<root url="jar://$USER_HOME$/.jdks/openjdk-17.0.1/lib/src.zip!/java.transaction.xa"
type="simple" />
<root url="jar://$USER_HOME$/.jdks/openjdk-17.0.1/lib/src.zip!/jdk.crypto.cryptoki"
type="simple" />
<root url="jar://$USER_HOME$/.jdks/openjdk-17.0.1/lib/src.zip!/jdk.incubator.vector"
type="simple" />
```

```
      <root url="jar://$USER_HOME$/.jdks/openjdk-17.0.1/lib/src.zip!/jdk.internal.jvmstat"
type="simple" />
      <root url="jar://$USER_HOME$/.jdks/openjdk-17.0.1/lib/src.zip!/jdk.management.agent"
type="simple" />
      <root url="jar://$USER_HOME$/.jdks/openjdk-17.0.1/lib/src.zip!/jdk.incubator.foreign"
type="simple" />
      <root url="jar://$USER_HOME$/.jdks/openjdk-17.0.1/lib/src.zip!/jdk.unsupported.desktop"
type="simple" />
      <root url="jar://$USER_HOME$/.jdks/openjdk-17.0.1/lib/src.zip!/jdk.internal.vm.compiler"
type="simple" />
      <root
url="jar://$USER_HOME$/.jdks/openjdk-17.0.1/lib/src.zip!/jdk.internal.vm.compiler.management"
type="simple" />
     </root>
    </sourcePath>
   </roots>
   <additional />
  </jdk>
  <jdk version="2">
   <name value="openjdk-17" />
   <type value="JavaSDK" />
   <version value="java version &quot;17.0.1&quot;" />
   <homePath value="$USER_HOME$/.jdks/openjdk-17.0.1" />
   <roots>
    <annotationsPath>
     <root type="composite">
```

```
                  <root  url="jar://$APPLICATION_HOME_DIR$/plugins/java/lib/jdkAnnotations.jar!/"
type="simple" />
     </root>
    </annotationsPath>
    <classPath>
     <root type="composite">
      <root url="jrt://C:/Users/user/.jdks/openjdk-17.0.1!/java.base" type="simple" />
      <root url="jrt://C:/Users/user/.jdks/openjdk-17.0.1!/java.compiler" type="simple" />
      <root url="jrt://C:/Users/user/.jdks/openjdk-17.0.1!/java.datatransfer" type="simple" />
      <root url="jrt://C:/Users/user/.jdks/openjdk-17.0.1!/java.desktop" type="simple" />
      <root url="jrt://C:/Users/user/.jdks/openjdk-17.0.1!/java.instrument" type="simple" />
      <root url="jrt://C:/Users/user/.jdks/openjdk-17.0.1!/java.logging" type="simple" />
      <root url="jrt://C:/Users/user/.jdks/openjdk-17.0.1!/java.management" type="simple" />
      <root url="jrt://C:/Users/user/.jdks/openjdk-17.0.1!/java.management.rmi" type="simple" />
      <root url="jrt://C:/Users/user/.jdks/openjdk-17.0.1!/java.naming" type="simple" />
      <root url="jrt://C:/Users/user/.jdks/openjdk-17.0.1!/java.net.http" type="simple" />
      <root url="jrt://C:/Users/user/.jdks/openjdk-17.0.1!/java.prefs" type="simple" />
      <root url="jrt://C:/Users/user/.jdks/openjdk-17.0.1!/java.rmi" type="simple" />
      <root url="jrt://C:/Users/user/.jdks/openjdk-17.0.1!/java.scripting" type="simple" />
      <root url="jrt://C:/Users/user/.jdks/openjdk-17.0.1!/java.se" type="simple" />
      <root url="jrt://C:/Users/user/.jdks/openjdk-17.0.1!/java.security.jgss" type="simple" />
      <root url="jrt://C:/Users/user/.jdks/openjdk-17.0.1!/java.security.sasl" type="simple" />
      <root url="jrt://C:/Users/user/.jdks/openjdk-17.0.1!/java.smartcardio" type="simple" />
      <root url="jrt://C:/Users/user/.jdks/openjdk-17.0.1!/java.sql" type="simple" />
      <root url="jrt://C:/Users/user/.jdks/openjdk-17.0.1!/java.sql.rowset" type="simple" />
      <root url="jrt://C:/Users/user/.jdks/openjdk-17.0.1!/java.transaction.xa" type="simple" />
```

```
        <root url="jrt://C:/Users/user/.jdks/openjdk-17.0.1!/java.xml" type="simple" />
        <root url="jrt://C:/Users/user/.jdks/openjdk-17.0.1!/java.xml.crypto" type="simple" />
        <root url="jrt://C:/Users/user/.jdks/openjdk-17.0.1!/jdk.accessibility" type="simple" />
        <root url="jrt://C:/Users/user/.jdks/openjdk-17.0.1!/jdk.attach" type="simple" />
        <root url="jrt://C:/Users/user/.jdks/openjdk-17.0.1!/jdk.charsets" type="simple" />
        <root url="jrt://C:/Users/user/.jdks/openjdk-17.0.1!/jdk.compiler" type="simple" />
        <root url="jrt://C:/Users/user/.jdks/openjdk-17.0.1!/jdk.crypto.cryptoki" type="simple" />
        <root url="jrt://C:/Users/user/.jdks/openjdk-17.0.1!/jdk.crypto.ec" type="simple" />
        <root url="jrt://C:/Users/user/.jdks/openjdk-17.0.1!/jdk.crypto.mscapi" type="simple" />
        <root url="jrt://C:/Users/user/.jdks/openjdk-17.0.1!/jdk.dynalink" type="simple" />
        <root url="jrt://C:/Users/user/.jdks/openjdk-17.0.1!/jdk.editpad" type="simple" />
        <root url="jrt://C:/Users/user/.jdks/openjdk-17.0.1!/jdk.hotspot.agent" type="simple" />
        <root url="jrt://C:/Users/user/.jdks/openjdk-17.0.1!/jdk.httpserver" type="simple" />
        <root url="jrt://C:/Users/user/.jdks/openjdk-17.0.1!/jdk.incubator.foreign" type="simple" />
        <root url="jrt://C:/Users/user/.jdks/openjdk-17.0.1!/jdk.incubator.vector" type="simple" />
        <root url="jrt://C:/Users/user/.jdks/openjdk-17.0.1!/jdk.internal.ed" type="simple" />
        <root url="jrt://C:/Users/user/.jdks/openjdk-17.0.1!/jdk.internal.jvmstat" type="simple" />
        <root url="jrt://C:/Users/user/.jdks/openjdk-17.0.1!/jdk.internal.le" type="simple" />
        <root url="jrt://C:/Users/user/.jdks/openjdk-17.0.1!/jdk.internal.opt" type="simple" />
        <root url="jrt://C:/Users/user/.jdks/openjdk-17.0.1!/jdk.internal.vm.ci" type="simple" />
        <root url="jrt://C:/Users/user/.jdks/openjdk-17.0.1!/jdk.internal.vm.compiler" type="simple" />
        <root url="jrt://C:/Users/user/.jdks/openjdk-17.0.1!/jdk.internal.vm.compiler.management" type="simple" />
        <root url="jrt://C:/Users/user/.jdks/openjdk-17.0.1!/jdk.jartool" type="simple" />
        <root url="jrt://C:/Users/user/.jdks/openjdk-17.0.1!/jdk.javadoc" type="simple" />
        <root url="jrt://C:/Users/user/.jdks/openjdk-17.0.1!/jdk.jcmd" type="simple" />
```

```
<root url="jrt://C:/Users/user/.jdks/openjdk-17.0.1!/jdk.jconsole" type="simple" />
<root url="jrt://C:/Users/user/.jdks/openjdk-17.0.1!/jdk.jdeps" type="simple" />
<root url="jrt://C:/Users/user/.jdks/openjdk-17.0.1!/jdk.jdi" type="simple" />
<root url="jrt://C:/Users/user/.jdks/openjdk-17.0.1!/jdk.jdwp.agent" type="simple" />
<root url="jrt://C:/Users/user/.jdks/openjdk-17.0.1!/jdk.jfr" type="simple" />
<root url="jrt://C:/Users/user/.jdks/openjdk-17.0.1!/jdk.jlink" type="simple" />
<root url="jrt://C:/Users/user/.jdks/openjdk-17.0.1!/jdk.jpackage" type="simple" />
<root url="jrt://C:/Users/user/.jdks/openjdk-17.0.1!/jdk.jshell" type="simple" />
<root url="jrt://C:/Users/user/.jdks/openjdk-17.0.1!/jdk.jsobject" type="simple" />
<root url="jrt://C:/Users/user/.jdks/openjdk-17.0.1!/jdk.jstatd" type="simple" />
<root url="jrt://C:/Users/user/.jdks/openjdk-17.0.1!/jdk.localedata" type="simple" />
<root url="jrt://C:/Users/user/.jdks/openjdk-17.0.1!/jdk.management" type="simple" />
<root url="jrt://C:/Users/user/.jdks/openjdk-17.0.1!/jdk.management.agent" type="simple" />
<root url="jrt://C:/Users/user/.jdks/openjdk-17.0.1!/jdk.management.jfr" type="simple" />
<root url="jrt://C:/Users/user/.jdks/openjdk-17.0.1!/jdk.naming.dns" type="simple" />
<root url="jrt://C:/Users/user/.jdks/openjdk-17.0.1!/jdk.naming.rmi" type="simple" />
<root url="jrt://C:/Users/user/.jdks/openjdk-17.0.1!/jdk.net" type="simple" />
<root url="jrt://C:/Users/user/.jdks/openjdk-17.0.1!/jdk.nio.mapmode" type="simple" />
<root url="jrt://C:/Users/user/.jdks/openjdk-17.0.1!/jdk.random" type="simple" />
<root url="jrt://C:/Users/user/.jdks/openjdk-17.0.1!/jdk.sctp" type="simple" />
<root url="jrt://C:/Users/user/.jdks/openjdk-17.0.1!/jdk.security.auth" type="simple" />
<root url="jrt://C:/Users/user/.jdks/openjdk-17.0.1!/jdk.security.jgss" type="simple" />
<root url="jrt://C:/Users/user/.jdks/openjdk-17.0.1!/jdk.unsupported" type="simple" />
<root url="jrt://C:/Users/user/.jdks/openjdk-17.0.1!/jdk.unsupported.desktop" type="simple" />
<root url="jrt://C:/Users/user/.jdks/openjdk-17.0.1!/jdk.xml.dom" type="simple" />
<root url="jrt://C:/Users/user/.jdks/openjdk-17.0.1!/jdk.zipfs" type="simple" />
```

```
  </root>
</classPath>
<javadocPath>
  <root type="composite" />
</javadocPath>
<sourcePath>
  <root type="composite">
    <root url="jar://$USER_HOME$/.jdks/openjdk-17.0.1/lib/src.zip!/java.se" type="simple" />
    <root url="jar://$USER_HOME$/.jdks/openjdk-17.0.1/lib/src.zip!/jdk.jdi" type="simple" />
    <root url="jar://$USER_HOME$/.jdks/openjdk-17.0.1/lib/src.zip!/jdk.jfr" type="simple" />
    <root url="jar://$USER_HOME$/.jdks/openjdk-17.0.1/lib/src.zip!/jdk.net" type="simple" />
    <root url="jar://$USER_HOME$/.jdks/openjdk-17.0.1/lib/src.zip!/java.rmi" type="simple" />
    <root url="jar://$USER_HOME$/.jdks/openjdk-17.0.1/lib/src.zip!/java.sql" type="simple" />
    <root url="jar://$USER_HOME$/.jdks/openjdk-17.0.1/lib/src.zip!/java.xml" type="simple" />
    <root url="jar://$USER_HOME$/.jdks/openjdk-17.0.1/lib/src.zip!/jdk.jcmd" type="simple" />
    <root url="jar://$USER_HOME$/.jdks/openjdk-17.0.1/lib/src.zip!/jdk.sctp" type="simple" />
    <root url="jar://$USER_HOME$/.jdks/openjdk-17.0.1/lib/src.zip!/java.base" type="simple" />
    <root url="jar://$USER_HOME$/.jdks/openjdk-17.0.1/lib/src.zip!/jdk.jdeps" type="simple" />
    <root url="jar://$USER_HOME$/.jdks/openjdk-17.0.1/lib/src.zip!/jdk.jlink" type="simple" />
    <root url="jar://$USER_HOME$/.jdks/openjdk-17.0.1/lib/src.zip!/jdk.zipfs" type="simple" />
    <root url="jar://$USER_HOME$/.jdks/openjdk-17.0.1/lib/src.zip!/java.prefs" type="simple" />
    <root url="jar://$USER_HOME$/.jdks/openjdk-17.0.1/lib/src.zip!/jdk.attach" type="simple" />
    <root url="jar://$USER_HOME$/.jdks/openjdk-17.0.1/lib/src.zip!/jdk.jshell" type="simple" />
    <root url="jar://$USER_HOME$/.jdks/openjdk-17.0.1/lib/src.zip!/jdk.jstatd" type="simple" />
    <root url="jar://$USER_HOME$/.jdks/openjdk-17.0.1/lib/src.zip!/jdk.random" type="simple" />
      <root url="jar://$USER_HOME$/.jdks/openjdk-17.0.1/lib/src.zip!/java.naming" type="simple"
```

```
/>
        <root url="jar://$USER_HOME$/.jdks/openjdk-17.0.1/lib/src.zip!/jdk.editpad" type="simple" />
        <root url="jar://$USER_HOME$/.jdks/openjdk-17.0.1/lib/src.zip!/jdk.jartool" type="simple" />
        <root url="jar://$USER_HOME$/.jdks/openjdk-17.0.1/lib/src.zip!/jdk.javadoc" type="simple"
/>
        <root url="jar://$USER_HOME$/.jdks/openjdk-17.0.1/lib/src.zip!/jdk.xml.dom" type="simple"
/>
        <root url="jar://$USER_HOME$/.jdks/openjdk-17.0.1/lib/src.zip!/java.desktop" type="simple"
/>
        <root url="jar://$USER_HOME$/.jdks/openjdk-17.0.1/lib/src.zip!/java.logging" type="simple"
/>
        <root url="jar://$USER_HOME$/.jdks/openjdk-17.0.1/lib/src.zip!/jdk.charsets" type="simple"
/>
        <root url="jar://$USER_HOME$/.jdks/openjdk-17.0.1/lib/src.zip!/jdk.compiler" type="simple"
/>
        <root url="jar://$USER_HOME$/.jdks/openjdk-17.0.1/lib/src.zip!/jdk.dynalink" type="simple"
/>
        <root url="jar://$USER_HOME$/.jdks/openjdk-17.0.1/lib/src.zip!/jdk.jconsole" type="simple"
/>
        <root url="jar://$USER_HOME$/.jdks/openjdk-17.0.1/lib/src.zip!/jdk.jpackage" type="simple"
/>
        <root url="jar://$USER_HOME$/.jdks/openjdk-17.0.1/lib/src.zip!/jdk.jsobject" type="simple" />
        <root url="jar://$USER_HOME$/.jdks/openjdk-17.0.1/lib/src.zip!/java.compiler" type="simple"
/>
        <root url="jar://$USER_HOME$/.jdks/openjdk-17.0.1/lib/src.zip!/java.net.http" type="simple"
/>
```

```
<root url="jar://$USER_HOME$/.jdks/openjdk-17.0.1/lib/src.zip!/jdk.crypto.ec" type="simple"
/>
<root url="jar://$USER_HOME$/.jdks/openjdk-17.0.1/lib/src.zip!/java.scripting" type="simple"
/>
<root url="jar://$USER_HOME$/.jdks/openjdk-17.0.1/lib/src.zip!/jdk.httpserver" type="simple"
/>
<root url="jar://$USER_HOME$/.jdks/openjdk-17.0.1/lib/src.zip!/jdk.jdwp.agent"
type="simple" />
<root url="jar://$USER_HOME$/.jdks/openjdk-17.0.1/lib/src.zip!/jdk.localedata"
type="simple" />
<root url="jar://$USER_HOME$/.jdks/openjdk-17.0.1/lib/src.zip!/jdk.management"
type="simple" />
<root url="jar://$USER_HOME$/.jdks/openjdk-17.0.1/lib/src.zip!/jdk.naming.dns"
type="simple" />
<root url="jar://$USER_HOME$/.jdks/openjdk-17.0.1/lib/src.zip!/jdk.naming.rmi"
type="simple" />
<root url="jar://$USER_HOME$/.jdks/openjdk-17.0.1/lib/src.zip!/java.instrument"
type="simple" />
<root url="jar://$USER_HOME$/.jdks/openjdk-17.0.1/lib/src.zip!/java.management"
type="simple" />
<root url="jar://$USER_HOME$/.jdks/openjdk-17.0.1/lib/src.zip!/java.sql.rowset"
type="simple" />
<root url="jar://$USER_HOME$/.jdks/openjdk-17.0.1/lib/src.zip!/java.xml.crypto"
type="simple" />
<root url="jar://$USER_HOME$/.jdks/openjdk-17.0.1/lib/src.zip!/jdk.internal.ed"
type="simple" />
```

```
<root url="jar://$USER_HOME$/.jdks/openjdk-17.0.1/lib/src.zip!/jdk.internal.le" type="simple"
/>
<root url="jar://$USER_HOME$/.jdks/openjdk-17.0.1/lib/src.zip!/jdk.nio.mapmode"
type="simple" />
<root url="jar://$USER_HOME$/.jdks/openjdk-17.0.1/lib/src.zip!/jdk.unsupported"
type="simple" />
<root url="jar://$USER_HOME$/.jdks/openjdk-17.0.1/lib/src.zip!/java.smartcardio"
type="simple" />
<root url="jar://$USER_HOME$/.jdks/openjdk-17.0.1/lib/src.zip!/jdk.internal.opt"
type="simple" />
<root url="jar://$USER_HOME$/.jdks/openjdk-17.0.1/lib/src.zip!/java.datatransfer"
type="simple" />
<root url="jar://$USER_HOME$/.jdks/openjdk-17.0.1/lib/src.zip!/jdk.accessibility"
type="simple" />
<root url="jar://$USER_HOME$/.jdks/openjdk-17.0.1/lib/src.zip!/jdk.crypto.mscapi"
type="simple" />
<root url="jar://$USER_HOME$/.jdks/openjdk-17.0.1/lib/src.zip!/jdk.hotspot.agent"
type="simple" />
<root url="jar://$USER_HOME$/.jdks/openjdk-17.0.1/lib/src.zip!/jdk.security.auth"
type="simple" />
<root url="jar://$USER_HOME$/.jdks/openjdk-17.0.1/lib/src.zip!/jdk.security.jgss"
type="simple" />
<root url="jar://$USER_HOME$/.jdks/openjdk-17.0.1/lib/src.zip!/java.security.jgss"
type="simple" />
<root url="jar://$USER_HOME$/.jdks/openjdk-17.0.1/lib/src.zip!/java.security.sasl"
type="simple" />
```

```
<root url="jar://$USER_HOME$/.jdks/openjdk-17.0.1/lib/src.zip!/jdk.internal.vm.ci"
type="simple" />
<root url="jar://$USER_HOME$/.jdks/openjdk-17.0.1/lib/src.zip!/jdk.management.jfr"
type="simple" />
<root url="jar://$USER_HOME$/.jdks/openjdk-17.0.1/lib/src.zip!/java.management.rmi"
type="simple" />
<root url="jar://$USER_HOME$/.jdks/openjdk-17.0.1/lib/src.zip!/java.transaction.xa"
type="simple" />
<root url="jar://$USER_HOME$/.jdks/openjdk-17.0.1/lib/src.zip!/jdk.crypto.cryptoki"
type="simple" />
<root url="jar://$USER_HOME$/.jdks/openjdk-17.0.1/lib/src.zip!/jdk.incubator.vector"
type="simple" />
<root url="jar://$USER_HOME$/.jdks/openjdk-17.0.1/lib/src.zip!/jdk.internal.jvmstat"
type="simple" />
<root url="jar://$USER_HOME$/.jdks/openjdk-17.0.1/lib/src.zip!/jdk.management.agent"
type="simple" />
<root url="jar://$USER_HOME$/.jdks/openjdk-17.0.1/lib/src.zip!/jdk.incubator.foreign"
type="simple" />
<root url="jar://$USER_HOME$/.jdks/openjdk-17.0.1/lib/src.zip!/jdk.unsupported.desktop"
type="simple" />
<root url="jar://$USER_HOME$/.jdks/openjdk-17.0.1/lib/src.zip!/jdk.internal.vm.compiler"
type="simple" />
<root
url="jar://$USER_HOME$/.jdks/openjdk-17.0.1/lib/src.zip!/jdk.internal.vm.compiler.management"
type="simple" />
    </root>
```

```
        </sourcePath>
      </roots>
      <additional />
    </jdk>
    <jdk version="2">
      <name value="jbr-11" />
      <type value="JavaSDK" />
      <version value="JetBrains Runtime version 11.0.13" />
      <homePath value="$APPLICATION_HOME_DIR$/jbr" />
      <roots>
        <annotationsPath>
          <root type="composite">
            <root url="jar://$APPLICATION_HOME_DIR$/plugins/java/lib/jdkAnnotations.jar!/" type="simple" />
          </root>
        </annotationsPath>
        <classPath>
          <root type="composite">
            <root url="jrt://C:/Program Files/JetBrains/IntelliJ IDEA 2021.3/jbr!/gluegen.rt" type="simple" />
            <root url="jrt://C:/Program Files/JetBrains/IntelliJ IDEA 2021.3/jbr!/java.base" type="simple" />
            <root url="jrt://C:/Program Files/JetBrains/IntelliJ IDEA 2021.3/jbr!/java.compiler" type="simple" />
            <root url="jrt://C:/Program Files/JetBrains/IntelliJ IDEA 2021.3/jbr!/java.datatransfer" type="simple" />
```

```
<root url="jrt://C:/Program Files/JetBrains/IntelliJ IDEA 2021.3/jbr!/java.desktop"
type="simple" />
<root url="jrt://C:/Program Files/JetBrains/IntelliJ IDEA 2021.3/jbr!/java.instrument"
type="simple" />
<root url="jrt://C:/Program Files/JetBrains/IntelliJ IDEA 2021.3/jbr!/java.logging"
type="simple" />
<root url="jrt://C:/Program Files/JetBrains/IntelliJ IDEA 2021.3/jbr!/java.management"
type="simple" />
<root url="jrt://C:/Program Files/JetBrains/IntelliJ IDEA 2021.3/jbr!/java.management.rmi"
type="simple" />
<root url="jrt://C:/Program Files/JetBrains/IntelliJ IDEA 2021.3/jbr!/java.naming"
type="simple" />
<root url="jrt://C:/Program Files/JetBrains/IntelliJ IDEA 2021.3/jbr!/java.net.http"
type="simple" />
<root url="jrt://C:/Program Files/JetBrains/IntelliJ IDEA 2021.3/jbr!/java.prefs" type="simple"
/>
<root url="jrt://C:/Program Files/JetBrains/IntelliJ IDEA 2021.3/jbr!/java.rmi" type="simple" />
<root url="jrt://C:/Program Files/JetBrains/IntelliJ IDEA 2021.3/jbr!/java.scripting"
type="simple" />
<root url="jrt://C:/Program Files/JetBrains/IntelliJ IDEA 2021.3/jbr!/java.se" type="simple" />
<root url="jrt://C:/Program Files/JetBrains/IntelliJ IDEA 2021.3/jbr!/java.security.jgss"
type="simple" />
<root url="jrt://C:/Program Files/JetBrains/IntelliJ IDEA 2021.3/jbr!/java.security.sasl"
type="simple" />
<root url="jrt://C:/Program Files/JetBrains/IntelliJ IDEA 2021.3/jbr!/java.smartcardio"
type="simple" />
```

```
<root url="jrt://C:/Program Files/JetBrains/IntelliJ IDEA 2021.3/jbr!/java.sql" type="simple" />
<root url="jrt://C:/Program Files/JetBrains/IntelliJ IDEA 2021.3/jbr!/java.sql.rowset" type="simple" />
<root url="jrt://C:/Program Files/JetBrains/IntelliJ IDEA 2021.3/jbr!/java.transaction.xa" type="simple" />
<root url="jrt://C:/Program Files/JetBrains/IntelliJ IDEA 2021.3/jbr!/java.xml" type="simple" />
<root url="jrt://C:/Program Files/JetBrains/IntelliJ IDEA 2021.3/jbr!/java.xml.crypto" type="simple" />
<root url="jrt://C:/Program Files/JetBrains/IntelliJ IDEA 2021.3/jbr!/jcef" type="simple" />
<root url="jrt://C:/Program Files/JetBrains/IntelliJ IDEA 2021.3/jbr!/jdk.accessibility" type="simple" />
<root url="jrt://C:/Program Files/JetBrains/IntelliJ IDEA 2021.3/jbr!/jdk.aot" type="simple" />
<root url="jrt://C:/Program Files/JetBrains/IntelliJ IDEA 2021.3/jbr!/jdk.attach" type="simple" />
<root url="jrt://C:/Program Files/JetBrains/IntelliJ IDEA 2021.3/jbr!/jdk.charsets" type="simple" />
<root url="jrt://C:/Program Files/JetBrains/IntelliJ IDEA 2021.3/jbr!/jdk.compiler" type="simple" />
<root url="jrt://C:/Program Files/JetBrains/IntelliJ IDEA 2021.3/jbr!/jdk.crypto.cryptoki" type="simple" />
<root url="jrt://C:/Program Files/JetBrains/IntelliJ IDEA 2021.3/jbr!/jdk.crypto.ec" type="simple" />
<root url="jrt://C:/Program Files/JetBrains/IntelliJ IDEA 2021.3/jbr!/jdk.dynalink" type="simple" />
<root url="jrt://C:/Program Files/JetBrains/IntelliJ IDEA 2021.3/jbr!/jdk.hotspot.agent" type="simple" />
```

```
<root url="jrt://C:/Program Files/JetBrains/IntelliJ IDEA 2021.3/jbr!/jdk.httpserver" type="simple" />
<root url="jrt://C:/Program Files/JetBrains/IntelliJ IDEA 2021.3/jbr!/jdk.internal.ed" type="simple" />
<root url="jrt://C:/Program Files/JetBrains/IntelliJ IDEA 2021.3/jbr!/jdk.internal.jvmstat" type="simple" />
<root url="jrt://C:/Program Files/JetBrains/IntelliJ IDEA 2021.3/jbr!/jdk.internal.le" type="simple" />
<root url="jrt://C:/Program Files/JetBrains/IntelliJ IDEA 2021.3/jbr!/jdk.internal.vm.ci" type="simple" />
<root url="jrt://C:/Program Files/JetBrains/IntelliJ IDEA 2021.3/jbr!/jdk.internal.vm.compiler" type="simple" />
<root url="jrt://C:/Program Files/JetBrains/IntelliJ IDEA 2021.3/jbr!/jdk.internal.vm.compiler.management" type="simple" />
<root url="jrt://C:/Program Files/JetBrains/IntelliJ IDEA 2021.3/jbr!/jdk.jcmd" type="simple" />
<root url="jrt://C:/Program Files/JetBrains/IntelliJ IDEA 2021.3/jbr!/jdk.jdi" type="simple" />
<root url="jrt://C:/Program Files/JetBrains/IntelliJ IDEA 2021.3/jbr!/jdk.jdwp.agent" type="simple" />
<root url="jrt://C:/Program Files/JetBrains/IntelliJ IDEA 2021.3/jbr!/jdk.jfr" type="simple" />
<root url="jrt://C:/Program Files/JetBrains/IntelliJ IDEA 2021.3/jbr!/jdk.jsobject" type="simple" />
<root url="jrt://C:/Program Files/JetBrains/IntelliJ IDEA 2021.3/jbr!/jdk.localedata" type="simple" />
<root url="jrt://C:/Program Files/JetBrains/IntelliJ IDEA 2021.3/jbr!/jdk.management" type="simple" />
<root url="jrt://C:/Program Files/JetBrains/IntelliJ IDEA 2021.3/jbr!/jdk.management.agent"
```

```
type="simple" />
          <root url="jrt://C:/Program Files/JetBrains/IntelliJ IDEA 2021.3/jbr!/jdk.management.jfr"
type="simple" />
          <root url="jrt://C:/Program Files/JetBrains/IntelliJ IDEA 2021.3/jbr!/jdk.naming.dns"
type="simple" />
          <root url="jrt://C:/Program Files/JetBrains/IntelliJ IDEA 2021.3/jbr!/jdk.naming.rmi"
type="simple" />
          <root url="jrt://C:/Program Files/JetBrains/IntelliJ IDEA 2021.3/jbr!/jdk.net" type="simple" />
          <root url="jrt://C:/Program Files/JetBrains/IntelliJ IDEA 2021.3/jbr!/jdk.pack" type="simple" />
          <root url="jrt://C:/Program Files/JetBrains/IntelliJ IDEA 2021.3/jbr!/jdk.scripting.nashorn"
type="simple" />
          <root url="jrt://C:/Program Files/JetBrains/IntelliJ IDEA
2021.3/jbr!/jdk.scripting.nashorn.shell" type="simple" />
          <root url="jrt://C:/Program Files/JetBrains/IntelliJ IDEA 2021.3/jbr!/jdk.sctp" type="simple" />
          <root url="jrt://C:/Program Files/JetBrains/IntelliJ IDEA 2021.3/jbr!/jdk.security.auth"
type="simple" />
          <root url="jrt://C:/Program Files/JetBrains/IntelliJ IDEA 2021.3/jbr!/jdk.security.jgss"
type="simple" />
          <root url="jrt://C:/Program Files/JetBrains/IntelliJ IDEA 2021.3/jbr!/jdk.unsupported"
type="simple" />
          <root url="jrt://C:/Program Files/JetBrains/IntelliJ IDEA 2021.3/jbr!/jdk.xml.dom"
type="simple" />
          <root url="jrt://C:/Program Files/JetBrains/IntelliJ IDEA 2021.3/jbr!/jdk.zipfs" type="simple" />
          <root url="jrt://C:/Program Files/JetBrains/IntelliJ IDEA 2021.3/jbr!/jogl.all" type="simple" />
        </root>
      </classPath>
```

```
        <javadocPath>
          <root type="composite">
            <root url="https://docs.oracle.com/en/java/javase/11/docs/api/" type="simple" />
          </root>
        </javadocPath>
        <sourcePath>
          <root type="composite" />
        </sourcePath>
      </roots>
      <additional />
    </jdk>
    <jdk version="2">
      <name value="Android API 32 Platform" />
      <type value="Android SDK" />
      <version value="JetBrains Runtime version 11.0.13" />
      <homePath value="$USER_HOME$/AppData/Local/Android/Sdk" />
      <roots>
        <annotationsPath>
          <root type="composite">
            <root url="jar://$USER_HOME$/AppData/Local/Android/Sdk/platforms/android-32/data/annotations.zip!/" type="simple" />
          </root>
        </annotationsPath>
        <classPath>
          <root type="composite">
```

```
            <root url="jar://$USER_HOME$/AppData/Local/Android/Sdk/platforms/android-32/android.jar!/" type="simple" />
            <root url="file://$USER_HOME$/AppData/Local/Android/Sdk/platforms/android-32/data/res" type="simple" />
          </root>
        </classPath>
        <javadocPath>
          <root type="composite">
            <root url="http://developer.android.com/reference/" type="simple" />
          </root>
        </javadocPath>
        <sourcePath>
          <root type="composite" />
        </sourcePath>
      </roots>
      <additional jdk="jbr-11" sdk="android-32" />
    </jdk>
  </component>
</application>
```

ASSISTANT

The XML snippet you've provided appears to be a configuration file from an Integrated Development Environment (IDE), most likely from JetBrains IntelliJ IDEA. This kind of configuration file is used by the IDE to manage project settings, particularly those related to the Java Development Kit (JDK) versions and settings.

Here's a breakdown of the key elements in this configuration:

1. **`<application>` and `<component>` Tags:**

   - These are the root elements of the configuration file. The `component` tag typically specifies a particular component of the IDE, in this case, `ProjectJdkTable`, which is likely related to the JDK settings for projects.

2. **`<jdk>` Tag:**

   - This tag defines a JDK configuration. The IDE can support multiple JDKs, and each `<jdk>` tag represents a different JDK setup.

3. **`<name>`, `<type>`, `<version>`, `<homePath>`:**

   - These tags provide details about the JDK:
     - `name`: The name of the JDK as recognized within the IDE.
     - `type`: The type of SDK, here it's `JavaSDK`.
     - `version`: The version of the JDK.
     - `homePath`: The path to the JDK installation on your system.

4. **`<roots>` Tag:**

   - This section defines various paths used by the JDK:
     - `annotationsPath`: Path to annotation libraries.
     - `classPath`: The classpath for the JDK, including paths to the runtime libraries.
     - `javadocPath`: Path to the JavaDoc documentation.
     - `sourcePath`: Path to the source files.

5. **`<root>` Tag:**

- Within each `roots` section, the `root` tag specifies individual paths, often pointing to specific JAR files or directories.

6. **Multiple JDK Entries:**

- The file includes configurations for different JDKs (like Oracle OpenJDK 17, JetBrains Runtime 11, Android SDK), each with its own settings.

This file is used by the IDE to understand where different parts of the JDK are located, which JDK to use for compiling and running applications, and where to find documentation and source code for reference. It's an essential part of the IDE's configuration, especially in a multi-JDK environment where you might be working with different versions of Java or different Java-based SDKs (like Android).

Modifying this file manually is generally not recommended unless you are sure about the changes, as incorrect configurations can lead to issues with project compilation and execution within the IDE.

USER

can i use it on my website

ASSISTANT

The XML configuration snippet you provided is specific to an Integrated Development Environment (IDE) like JetBrains IntelliJ IDEA and is used for managing Java Development Kit (JDK) settings within the IDE. It is not directly applicable or usable for a website in the traditional sense. Here's why:

1. **IDE-Specific Configuration:** This XML file is used by the IDE to manage project settings, particularly for Java projects. It tells the IDE which JDK to use, where to find the JDK files, and other Java-specific settings.

2. **Not Web Content:** The content of this XML file does not represent web content (like HTML, CSS, or JavaScript), nor does it include any server-side code (like PHP, Python, or Node.js) that can be executed to serve web pages.

3. **No Direct Role in Web Development:** While Java can be used in web development (e.g., Java servlets, JSPs, or backend services), this specific XML configuration does not directly translate into any web functionality. It's more about setting up the development environment rather than creating content or functionality for a website.

4. **Security and Relevance:** Even if there were a way to use it, exposing your IDE's configuration settings on a public website is not advisable for security reasons. Moreover, it would not provide any meaningful interaction or value to the users of your website.

For building a website, especially one related to Quantum Bank AI as mentioned in your previous queries, you would typically use a combination of HTML, CSS, JavaScript for the frontend, and possibly a server-side language like Python, Node.js, or Java for backend processing. The website would also involve database interactions, API integrations, and security implementations, none of which would be facilitated by the IDE configuration file you've mentioned.

If you're looking to develop a website and need assistance with the actual code or architecture relevant to web development, feel free to ask! I can help guide you through creating web pages, setting up server-side scripts, and more.

USER

well how do i make my website an ide

ASSISTANT

Creating a web-based Integrated Development Environment (IDE) is a complex task that involves several key components. A web-based IDE allows users to write, edit, compile, and possibly run code directly in a web browser. Here's a high-level overview of what you would need to consider:

1. **Frontend Development:**

- **Code Editor:** Implement a code editor in the browser. Libraries like [Ace](https://ace.c9.io/) or [CodeMirror](https://codemirror.net/) can be used to embed a syntax-highlighting editor into your web page.
- **User Interface:** Design a user-friendly interface that includes areas for the code editor, file explorer, console/output area, and possibly debugging tools.

2. **Backend Development:**

- **Language Support:** Decide which programming languages your IDE will support. Each language may require different compilers or interpreters.
- **Compilation and Execution:** Implement a secure way to compile and run user code. This often involves sandboxing the execution to prevent security vulnerabilities.
- **File Management:** Implement a system for users to manage their files and projects, including saving, retrieving, and organizing files.

3. **Integration of Frontend and Backend:**

- **APIs:** Develop APIs to handle requests between the frontend and backend, such as compiling code, executing code, and managing files.

- **Real-Time Interaction:** Optionally, you can implement real-time collaboration features (like in Google Docs).

4. **Security Considerations:**

- **Execution Security:** Running user code on the server can be risky. Sandboxing, resource limits, and other security measures are crucial.

- **Authentication:** Implement user authentication to enable personalized workspaces and secure access.

5. **Additional Features:**

- **Debugging Tools:** Advanced IDEs include debugging tools. Implementing these can be quite complex.

- **Extensions and Plugins:** Consider allowing extensions or plugins for additional functionalities.

Example Architecture:

- **Frontend:** HTML/CSS/JavaScript with a framework like React or Vue.js.

- **Backend:** Node.js or Python Flask/Django for handling API requests.

- **Execution Environment:** Docker containers or a similar technology for safely executing user code.

- **Database:** For storing user data, project files, etc. (e.g., MongoDB, PostgreSQL).

Steps to Get Started:

1. **Prototype the Basic IDE:** Start with a simple code editor in the browser using Ace or CodeMirror.
2. **Set Up a Basic Backend:** Create a server that can receive code, compile it, and return the results. Start with one programming language.
3. **Implement File Management:** Allow users to save and retrieve their files.
4. **Iteratively Add Features:** Gradually add more languages, debugging tools, and user management features.

Considerations:

- **Complexity:** Building a full-featured web-based IDE is a significant project, typically requiring a team of developers.
- **Security:** Running arbitrary code is inherently risky; ensure robust security measures.
- **Performance:** Consider the resource requirements for compiling and running user code.

Building a web-based IDE from scratch is a large and complex task, often requiring a team of experienced developers. If you're new to this, you might want to start small, perhaps by creating a simple code editor, and gradually add features as you learn more.

USER

1
SECURITIES AND EXCHANGE COMMISSION
SECURITIES EXCHANGE ACT OF 1934
Release No. 81207 / July 25, 2017
Report of Investigation Pursuant to Section 21(a) of the Securities Exchange Act of 1934:
The DAO

I. Introduction and Summary

The United States Securities and Exchange Commission's ("Commission") Division of Enforcement ("Division") has investigated whether The DAO, an unincorporated organization; Slock.it UG ("Slock.it"), a German corporation; Slock.it's co-founders; and intermediaries may have violated the federal securities laws. The Commission has determined not to pursue an enforcement action in this matter based on the conduct and activities known to the Commission at this time.

As described more fully below, The DAO is one example of a Decentralized Autonomous Organization, which is a term used to describe a "virtual" organization embodied in computer code and executed on a distributed ledger or blockchain. The DAO was created by Slock.it and Slock.it's co-founders, with the objective of operating as a for-profit entity that would create and hold a corpus of assets through the sale of DAO Tokens to investors, which assets would then be used to fund "projects." The holders of DAO Tokens stood to share in the anticipated earnings from these projects as a return on their investment in DAO Tokens. In addition, DAO Token holders could monetize their investments in DAO Tokens by re-selling DAO Tokens on a number of web-based platforms ("Platforms") that supported secondary trading in the DAO Tokens.

After DAO Tokens were sold, but before The DAO was able to commence funding projects, an attacker used a flaw in The DAO's code to steal approximately one-third of The DAO's assets. Slock.it's co-founders and others responded by creating a work-around whereby DAO Token holders could opt to have their investment returned to them, as described in more detail below.

The investigation raised questions regarding the application of the U.S. federal securities laws to the offer and sale of DAO Tokens, including the threshold question whether DAO Tokens are securities. Based on the investigation, and under the facts presented, the Commission has determined that DAO Tokens are securities under the Securities Act of 1933 ("Securities

Act") and the Securities Exchange Act of 1934 ("Exchange Act").

1

The Commission deems it appropriate and in the public interest to issue this report of investigation ("Report") pursuant to

1

This Report does not analyze the question whether The DAO was an "investment company," as defined under Section 3(a) of the Investment Company Act of 1940 ("Investment Company Act"), in part, because The DAO never commenced its business operations funding projects. Those who would use virtual organizations should consider their obligations under the Investment Company Act.

2

Section 21(a) of the Exchange Act2 to advise those who would use a Decentralized Autonomous Organization ("DAO Entity"), or other distributed ledger or blockchain-enabled means for capital raising, to take appropriate steps to ensure compliance with the U.S. federal securities laws. All securities offered and sold in the United States must be registered with the Commission or must qualify for an exemption from the registration requirements. In addition, any entity or person engaging in the activities of an exchange must register as a national securities exchange or operate pursuant to an exemption from such registration.

This Report reiterates these fundamental principles of the U.S. federal securities laws and describes their applicability to a new paradigm?virtual organizations or capital raising entities that use distributed ledger or blockchain technology to facilitate capital raising and/or investment and the related offer and sale of securities. The automation of certain functions through this technology, "smart contracts,"3 or computer code, does not remove conduct from the purview of the U.S. federal securities laws.4 This Report also serves to stress the obligation to comply with the registration provisions of the federal securities laws with respect to products and platforms involving emerging technologies and new investor interfaces.

II. Facts

A. Background

From April 30, 2016 through May 28, 2016, The DAO offered and sold approximately 1.15 billion DAO Tokens in exchange for a total of approximately 12 million Ether ("ETH"), a

[2] Section 21(a) of the Exchange Act authorizes the Commission to investigate violations of the federal securities laws and, in its discretion, to "publish information concerning any such violations." This Report does not constitute an adjudication of any fact or issue addressed herein, nor does it make any findings of violations by any individual or entity. The facts discussed in Section II, infra, are matters of public record or based on documentary records. We are publishing this Report on the Commission's website to ensure that all market participants have concurrent and equal access to the information contained herein.

[3] Computer scientist Nick Szabo described a "smart contract" as:

a computerized transaction protocol that executes terms of a contract. The general objectives of smart contract design are to satisfy common contractual conditions (such as payment terms, liens, confidentiality, and even enforcement), minimize exceptions both malicious and accidental, and minimize the need for trusted intermediaries. Related economic goals include lowering fraud loss, arbitrations and enforcement costs, and other transaction costs.

See Nick Szabo, Smart Contracts, 1994, http://www.virtualschool.edu/mon/Economics/SmartContracts.html.

[4] See SEC v. C.M. Joiner Leasing Corp., 320 U.S. 344, 351 (1943) ("[T]he reach of the [Securities] Act does not stop with the obvious and commonplace. Novel, uncommon, or irregular devices, whatever they appear to be, are also reached if it be proved as matter of fact that they were widely offered or dealt in under terms or courses of dealing which established their character in commerce as ?investment contracts,' or as ?any interest or instrument commonly known as a ?security'."); see also Reves v. Ernst & Young, 494 U.S. 56, 61 (1990) ("Congress' purpose in enacting the securities laws was to regulate investments, in whatever form they are made and by whatever name they are called.").

virtual currency[5] used on the Ethereum Blockchain.[6] As of the time the offering closed, the total ETH raised by The DAO was valued in U.S. Dollars ("USD") at approximately $150 million.

The concept of a DAO Entity is memorialized in a document (the "White Paper"), authored by Christoph Jentzsch, the Chief Technology Officer of Slock.it, a "Blockchain and IoT [(internet-of-things)] solution company," incorporated in Germany and co-founded by Christoph Jentzsch, Simon Jentzsch (Christoph Jentzsch's brother), and Stephan Tual ("Tual").[7] The White Paper purports to describe "the first implementation of a [DAO Entity] code to automate organizational governance and decision making."[8] The White Paper posits that a DAO Entity "can be used by individuals working together collaboratively outside of a traditional corporate form. It can also be used by a registered corporate entity to automate formal governance rules contained in corporate bylaws or imposed by law." The White Paper proposes an entity?a DAO Entity?that would use smart contracts to attempt to solve governance issues it described as inherent in traditional corporations.9

As described, a DAO Entity purportedly would supplant traditional mechanisms of corporate governance and management with a blockchain such that contractual terms are "formalized, automated and enforced using software."10

5

The Financial Action Task Force defines "virtual currency" as:

a digital representation of value that can be digitally traded and functions as: (1) a medium of exchange; and/or (2) a unit of account; and/or (3) a store of value, but does not have legal tender status (i.e., when tendered to a creditor, is a valid and legal offer of payment) in any jurisdiction. It is not issued or guaranteed by any jurisdiction, and fulfils the above functions only by agreement within the community of users of the virtual currency. Virtual currency is distinguished from fiat currency (a.k.a. "real currency," "real money," or "national currency"), which is the coin and paper money of a country that is designated as its legal tender; circulates; and is customarily used and accepted as a medium of exchange in the issuing country. It is distinct from e-money, which is a digital representation of fiat currency used to electronically transfer value denominated in fiat currency.

FATF Report, Virtual Currencies, Key Definitions and Potential AML/CFT Risks, FINANCIAL ACTION TASK FORCE (June 2014), http://www.fatf-gafi.org/media/fatf/documents/reports/Virtual-currency-key-definitions-and-potentiala ml-cft-risks.pdf.

6

Ethereum, developed by the Ethereum Foundation, a Swiss nonprofit organization, is a decentralized platform that runs smart contracts on a blockchain known as the Ethereum Blockchain.

7

Christoph Jentzsch released the final draft of the White Paper on or around March 23, 2016. He introduced his concept of a DAO Entity as early as November 2015 at an Ethereum Developer Conference in London, as a medium to raise funds for Slock.it, a German start-up he co-founded in September 2015. Slock.it purports to create technology that embeds smart contracts that run on the Ethereum Blockchain into real-world devices and, as a result, for example, permits anyone to rent, sell or share physical objects in a decentralized way. See SLOCK.IT, https://slock.it/.

8

Christoph Jentzsch, Decentralized Autonomous Organization to Automate Governance Final Draft ? Under Review, https://download.slock.it/public/DAO/WhitePaper.pdf.

9

Id.

10 Id. The White Paper contained the following statement:

A word of caution, at the outset: the legal status of [DAO Entities] remains the subject of active and vigorous debate and discussion. Not everyone shares the same definition. Some have said that [DAO Entities] are autonomous code and can operate independently of legal systems; others

B. The DAO

"The DAO" is the "first generation" implementation of the White Paper concept of a DAO Entity, and it began as an effort to create a "crowdfunding contract" to raise "funds to grow [a] company in the crypto space."11 In November 2015, at an Ethereum Developer Conference in London, Christoph Jentzsch described his proposal for The DAO as a "for-profit DAO [Entity]," where participants would send ETH (a virtual currency) to The DAO to purchase DAO Tokens, which would permit the participant to vote and entitle the participant to "rewards."12 Christoph Jentzsch likened this to "buying shares in a company and getting ? dividends."13 The DAO was to be "decentralized" in that it would allow for voting by investors holding DAO Tokens.

14 All funds raised were to be held at an Ethereum Blockchain "address" associated with The DAO and DAO Token holders were to vote on contract proposals, including proposals to The DAO to fund projects and distribute The DAO's anticipated earnings from the projects it funded.

15 The DAO was intended to be "autonomous" in that project proposals were in the form of smart contracts that exist on the Ethereum Blockchain and the votes were administered by the code of The DAO.

16

have said that [DAO Entities] must be owned or operate[d] by humans or human created entities. There will be many use cases, and the DAO [Entity] code will develop over time. Ultimately, how a DAO [Entity] functions and its legal status will depend on many factors, including how DAO [Entity] code is used, where it is used, and who uses it. This paper does not speculate about the legal status of [DAO Entities] worldwide. This paper is not intended to offer legal advice or conclusions. Anyone who uses DAO [Entity] code will do so at their own risk.

Id.

11 Christoph Jentzsch, The History of the DAO and Lessons Learned, SLOCK.IT BLOG (Aug. 24, 2016), https://blog.slock.it/the-history-of-the-dao-and-lessons-learned-d06740f8cfa5#.5o62zo8uv. Although The DAO has been described as a "crowdfunding contract," The DAO would not have met the requirements of Regulation Crowdfunding, adopted under Title III of the Jumpstart Our Business Startups (JOBS) Act of 2012 (providing an exemption from registration for certain crowdfunding), because, among other things, it was not a broker-dealer or a funding portal registered with the SEC and the Financial Industry Regulatory Authority ("FINRA"). See Regulation Crowdfunding: A Small Entity Compliance Guide for Issuers, SEC (Apr. 5, 2017), https://www.sec.gov/info/smallbus/secg/rccomplianceguide-051316.htm; Updated Investor Bulletin: Crowdfunding for Investors, SEC (May 10, 2017), https://www.sec.gov/oiea/investor-alerts-bulletins/ib_crowdfunding-.html.

12 See Slockit, Slock.it DAO demo at Devcon1: IoT + Blockchain, YOUTUBE (Nov. 13, 2015), https://www.youtube.com/watch?v=49wHQoJxYPo.

13 Id.

14 See Jentzsch, supra note 8.

15 Id. In theory, there was no limitation on the type of project that could be proposed. For example, proposed "projects" could include, among other things, projects that would culminate in the creation of products or services that DAO Token holders could use or charge others for using.

16 Id.

On or about April 29, 2016, Slock.it deployed The DAO code on the Ethereum Blockchain, as a set of pre-programmed instructions.

17 This code was to govern how The DAO was to operate.

To promote The DAO, Slock.it's co-founders launched a website ("The DAO Website").

The DAO Website included a description of The DAO's intended purpose: "To blaze a new path in business for the betterment of its members, existing simultaneously nowhere and everywhere and operating solely with the steadfast iron will of unstoppable code."18 The DAO Website also described how The DAO operated, and included a link through which DAO Tokens could be purchased. The DAO Website also included a link to the White Paper, which provided detailed information about a DAO Entity's structure and its source code and, together with The DAO Website, served as the primary source of promotional materials for The DAO. On The DAO Website and elsewhere, Slock.it represented that The DAO's source code had been reviewed by "one of the world's leading security audit companies" and "no stone was left unturned during those five whole days of security analysis."19

Slock.it's co-founders also promoted The DAO by soliciting media attention and by posting almost daily updates on The DAO's status on The DAO and Slock.it websites and numerous online forums relating to blockchain technology. Slock.it's co-founders used these posts to communicate to the public information about how to participate in The DAO, including: how to create and acquire DAO Tokens; the framework for submitting proposals for projects; and how to vote on proposals. Slock.it also created an online forum on The DAO Website, as well as administered "The DAO Slack" channel, an online messaging platform in which over 5,000 invited "team members" could discuss and exchange ideas about The DAO in real time.

1. DAO Tokens

In exchange for ETH, The DAO created DAO Tokens (proportional to the amount of ETH paid) that were then assigned to the Ethereum Blockchain address of the person or entity remitting the ETH. A DAO Token granted the DAO Token holder certain voting and ownership rights. According to promotional materials, The DAO would earn profits by funding projects

17 According to the White Paper, a DAO Entity is "activated by deployment on the Ethereum [B]lockchain. Once deployed, a [DAO Entity's] code requires ?ether' [ETH] to engage in transactions on Ethereum. Ether is the digital fuel that powers the Ethereum Network." The only way to update or alter The DAO's code is to submit a new proposal for voting and achieve a majority consensus on that proposal. See Jentzsch, supra note 8. According to Slock.it's website, Slock.it gave The DAO code to the Ethereum community, noting that:

> The DAO framework is [a] side project of Slock.it UG and a gift to the Ethereum community. It consisted of a definitive whitepaper, smart contract code audited by one of the best security companies in the world and soon, a complete frontend interface. All free and open source for anyone to re-use, it is our way to say ?thank you' to the community.

SLOCK.IT, https://slock.it. The DAO code is publicly-available on GitHub, a host of source code. See The Standard DAO Framework, Inc., Whitepaper, GITHUB, https://github.com/slockit/DAO.

18 The DAO Website was available at https://daohub.org.

19 Stephen Tual, Deja Vu DAO Smart Contracts Audit Results, SLOCK.IT BLOG (Apr. 5, 2016), https://blog.slock.it/deja-vu-dai-smart-contracts-audit-results-d26bc088e32e.

that would provide DAO Token holders a return on investment. The various promotional materials disseminated by Slock.it's co-founders touted that DAO Token holders would receive "rewards," which the White Paper defined as, "any [ETH] received by a DAO [Entity] generated from projects the DAO [Entity] funded." DAO Token holders would then vote to either use the rewards to fund new projects or to distribute the ETH to DAO Token holders.

From April 30, 2016 through May 28, 2016 (the "Offering Period"), The DAO offered and sold DAO Tokens. Investments in The DAO were made "pseudonymously" (i.e., an individual's or entity's pseudonym was their Ethereum Blockchain address). To purchase a DAO Token offered for sale by The DAO, an individual or entity sent ETH from their Ethereum Blockchain address to an Ethereum Blockchain address associated with The DAO. All of the ETH raised in the offering as well as any future profits earned by The DAO were to be pooled and held in The DAO's Ethereum Blockchain address. The token price fluctuated in a range of approximately 1 to 1.5 ETH per 100 DAO Tokens, depending on when the tokens were purchased during the Offering Period. Anyone was eligible to purchase DAO Tokens (as long as they paid ETH). There were no limitations placed on the number of DAO Tokens offered for sale, the number of purchasers of DAO Tokens, or the level of sophistication of such purchasers. DAO Token holders were not restricted from re-selling DAO Tokens acquired in the offering, and DAO Token holders could sell their DAO Tokens in a variety of ways in the secondary market and thereby monetize their investment as discussed below. Prior to the Offering Period, Slock.it solicited at least one U.S. web-based platform to trade DAO Tokens on its system and, at the time of the offering, The DAO Website and other promotional materials disseminated by Slock.it included representations that DAO Tokens would be available for secondary market trading after the Offering Period via several platforms. During the Offering Period and afterwards, the Platforms posted notices on their own websites and on social media that each planned to support secondary market trading of DAO Tokens.20

In addition to secondary market trading on the Platforms, after the Offering Period, DAO Tokens were to be freely transferable on the Ethereum Blockchain. DAO Token holders would also be permitted to redeem their DAO Tokens for ETH through a complicated, multi-week (approximately 46-day) process referred to as a DAO Entity "split."[21]

2. Participants in The DAO

According to the White Paper, in order for a project to be considered for funding with "a DAO [Entity]'s [ETH]," a "Contractor" first must submit a proposal to the DAO Entity. Specifically, DAO Token holders expected Contractors to submit proposals for projects that could provide DAO Token holders returns on their investments. Submitting a proposal to The DAO involved: (1) writing a smart contract, and then deploying and publishing it on the

[20] The Platforms are registered with FinCEN as "Money Services Businesses" and provide systems whereby customers may exchange virtual currencies for other virtual currencies or fiat currencies.

[21] According to the White Paper, the primary purpose of a split is to protect minority shareholders and prevent what is commonly referred to as a "51% Attack," whereby an attacker holding 51% of a DAO Entity's Tokens could create a proposal to send all of the DAO Entity's funds to himself or herself.

Ethereum Blockchain; and (2) posting details about the proposal on The DAO Website, including the Ethereum Blockchain address of the deployed contract and a link to its source code. Proposals could be viewed on The DAO Website as well as other publicly-accessible websites. Per the White Paper, there were two prerequisites for submitting a proposal. An individual or entity must: (1) own at least one DAO Token; and (2) pay a deposit in the form of ETH that would be forfeited to the DAO Entity if the proposal was put up for a vote and failed to achieve a quorum of DAO Token holders. It was publicized that Slock.it would be the first to submit a proposal for funding.[22]

ETH raised by The DAO was to be distributed to a Contractor to fund a proposal only on a majority vote of DAO Token holders.[23] DAO Token holders were to cast votes, which would be weighted by the number of tokens they controlled, for or against the funding of a specific proposal. The voting process, however, was publicly criticized in that it could incentivize distorted voting behavior and, as a result, would not accurately reflect the consensus of the majority of DAO Token holders. Specifically, as noted in a May 27, 2016 blog post by a group of computer security researchers, The DAO's structure included a "strong positive bias to vote YES on proposals and to suppress NO votes as a side effect of the way in which it restricts users' range of options following the casting of a vote."24

Before any proposal was put to a vote by DAO Token holders, it was required to be reviewed by one or more of The DAO's "Curators." At the time of the formation of The DAO, the Curators were a group of individuals chosen by Slock.it.25 According to the White Paper, the Curators of a DAO Entity had "considerable power." The Curators performed crucial security functions and maintained ultimate control over which proposals could be submitted to, voted on, and funded by The DAO. As stated on The DAO Website during the Offering Period, The DAO relied on its Curators for "failsafe protection" and for protecting The DAO from "malicous [sic] actors." Specifically, per The DAO Website, a Curator was responsible for: (1) confirming that any proposal for funding originated from an identifiable person or organization; and (2)

22 It was stated on The DAO Website and elsewhere that Slock.it anticipated that it would be the first to submit a proposal for funding. In fact, a draft of Slock.it's proposal for funding for an "Ethereum Computer and Universal Sharing Network" was publicly-available online during the Offering Period.

23 DAO Token holders could vote on proposals, either by direct interaction with the Ethereum Blockchain or by using an application that interfaces with the Ethereum Blockchain. It was generally acknowledged that DAO Token holders needed some technical knowledge in order to submit a vote, and The DAO Website

included a link to a stepby-step tutorial describing how to vote on proposals.

24 By voting on a proposal, DAO Token holders would "tie up" their tokens until the end of the voting cycle. See

Jentzsch, supra note 8 at 8 ("The tokens used to vote will be blocked, meaning they can not [sic] be transferred until

the proposal is closed."). If, however, a DAO Token holder abstained from voting, the DAO Token holder could

avoid these restrictions; any DAO Tokens not submitted for a vote could be withdrawn or transferred at any time.

As a result, DAO Token holders were incentivized either to vote yes or to abstain from voting. See Dino Mark et al.,

A Call for a Temporary Moratorium on The DAO, HACKING, DISTRIBUTED (May 27, 2016, 1:35 PM),

http://hackingdistributed.com/2016/05/27/dao-call-for-moratorium/.

25 At the time of The DAO's launch, The DAO Website identified eleven "high profile" individuals as holders of

The DAO's Curator "Multisig" (or "private key"). These individuals all appear to live outside of the United States.

Many of them were associated with the Ethereum Foundation, and The DAO Website touted the qualifications and

trustworthiness of these individuals.

confirming that smart contracts associated with any such proposal properly reflected the code the Contractor claims to have deployed on the Ethereum Blockchain. If a Curator determined that the proposal met these criteria, the Curator could add the proposal to the "whitelist," which was a list of Ethereum Blockchain addresses that could receive ETH from The DAO if the majority of

DAO Token holders voted for the proposal.

Curators of The DAO had ultimate discretion as to whether or not to submit a proposal for voting by DAO Token holders. Curators also determined the order and frequency of proposals, and could impose subjective criteria for whether the proposal should be whitelisted. One member of the group chosen by Slock.it to serve collectively as the Curator stated publicly that the Curator had "complete control over the whitelist ? the order in which things get whitelisted, the duration for which [proposals] get whitelisted, when things get unwhitelisted ? [and] clear ability to control the order and frequency of proposals," noting that "curators have tremendous power."26 Another Curator publicly announced his subjective criteria for determining whether to whitelist a proposal, which included his personal ethics.27 Per the White Paper, a Curator also had the power to reduce the voting quorum requirement by 50% every other week. Absent action by a Curator, the quorum could be reduced by 50% only if no proposal had reached the required quorum for 52 weeks.

3. Secondary Market Trading on the Platforms

During the period from May 28, 2016 through early September 2016, the Platforms became the preferred vehicle for DAO Token holders to buy and sell DAO Tokens in the secondary market using virtual or fiat currencies. Specifically, the Platforms used electronic systems that allowed their respective customers to post orders for DAO Tokens on an anonymous basis. For example, customers of each Platform could buy or sell DAO Tokens by entering a market order on the Platform's system, which would then match with orders from other customers residing on the system. Each Platform's system would automatically execute these orders based on pre-programmed order interaction protocols established by the Platform. None of the Platforms received orders for DAO Tokens from non-Platform customers or routed its respective customers' orders to any other trading destinations. The Platforms publicly displayed all their quotes, trades, and daily trading volume in DAO Tokens on their respective websites. During the period from May 28, 2016 through September 6, 2016, one such Platform executed more than 557,378 buy and sell transactions in DAO Tokens by more than 15,000 of its U.S. and foreign customers. During the period from May 28, 2016 through August 1, 2016, another such Platform executed more than 22,207 buy and sell transactions in DAO Tokens by more than 700 of its U.S. customers.

26 Epicenter, EB134 ? Emin Gün Sirer And Vlad Zamfir: On A Rocky DAO, YOUTUBE (June 6, 2016),

https://www.youtube.com/watch?v=ON5GhIQdFU8.

27 Andrew Quentson, Are the DAO Curators Masters or Janitors?, THE COIN TELEGRAPH (June 12, 2016),

https://cointelegraph.com/news/are-the-dao-curators-masters-or-janitors.

4. Security Concerns, The "Attack" on The DAO, and The Hard Fork

In late May 2016, just prior to the expiration of the Offering Period, concerns about the safety and security of The DAO's funds began to surface due to vulnerabilities in The DAO's code. On May 26, 2016, in response to these concerns, Slock.it submitted a "DAO Security Proposal" that called for the development of certain updates to The DAO's code and the appointment of a security expert.28 Further, on June 3, 2016, Christoph Jentzsch, on behalf of Slock.it, proposed a moratorium on all proposals until alterations to The DAO's code to fix vulnerabilities in The DAO's code had been implemented.29

On June 17, 2016, an unknown individual or group (the "Attacker") began rapidly diverting ETH from The DAO, causing approximately 3.6 million ETH?1/3 of the total ETH raised by The DAO offering?to move from The DAO's Ethereum Blockchain address to an Ethereum Blockchain address controlled by the Attacker (the "Attack").

30 Although the diverted

ETH was then held in an address controlled by the Attacker, the Attacker was prevented by The DAO's code from moving the ETH from that address for 27 days.

In order to secure the diverted ETH and return it to DAO Token holders, Slock.it's cofounders and others endorsed a "Hard Fork" to the Ethereum Blockchain. The "Hard Fork," called for a change in the Ethereum protocol on a going forward basis that would restore the DAO Token holders' investments as if the Attack had not occurred. On July 20, 2016, after a majority of the Ethereum network adopted the necessary software updates, the new, forked Ethereum Blockchain became active.[32] The Hard Fork had the effect of transferring all of the funds raised (including those held by the Attacker) from The DAO to a recovery address, where DAO Token holders could exchange their DAO Tokens for ETH.[33] All DAO Token holders

[28] See Stephan Tual, Proposal #1-DAO Security, Redux, SLOCK.IT BLOG (May 26, 2016), https://blog.slock.it/bothour-proposals-are-now-out-voting-starts-saturday-morning-ba322d6d3aea. The unnamed security expert would "act as the first point of contact for security disclosures, and continually monitor, pre-empt and avert any potential attack vectors The DAO may face, including social, technical and economic attacks." Id. Slock.it initially proposed a much broader security proposal that included the formation of a "DAO Security" group, the establishment of a "Bug Bounty Program," and routine external audits of The DAO's code. However, the cost of the proposal (125,000 ETH), which would be paid from The DAO's funds, was immediately criticized as too high and Slock.it decided instead to submit the revised proposal described above. See Stephan Tual, DAO.Security, a Proposal to guarantee the integrity of The DAO, SLOCK.IT BLOG (May 25, 2016), https://blog.slock.it/dao-security-a-proposal-toguarantee-the-integrity-of-the-dao-3473899ace9d.

[29] See TheDAO Proposal_ID 5, ETHERSCAN, https://etherscan.io/token/thedao-proposal/5.

[30] See Stephan Tual, DAO Security Advisory: live updates, SLOCK.IT BLOG (June 17, 2016), https://blog.slock.it/daosecurity-advisory-live-updates-2a0a42a2d07b.

[31] Id.

[32] A minority group, however, elected not to adopt the new Ethereum Blockchain created by the Hard Fork because to do so would run counter to the concept that a blockchain is immutable. Instead they continued to use the former version of the blockchain, which is now known as "Ethereum Classic."

[33] See Christoph Jentzsch, What the ?Fork' Really Means, SLOCK.IT BLOG (July 18, 2016), https://blog.slock.it/whatthe-fork-really-means-6fe573ac31dd.

who adopted the Hard Fork could exchange their DAO Tokens for ETH, and avoid any loss of the ETH they had invested.

34

III. Discussion

The Commission is aware that virtual organizations and associated individuals and entities increasingly are using distributed ledger technology to offer and sell instruments such as DAO Tokens to raise capital. These offers and sales have been referred to, among other things, as "Initial Coin Offerings" or "Token Sales." Accordingly, the Commission deems it appropriate and in the public interest to issue this Report in order to stress that the U.S. federal securities law may apply to various activities, including distributed ledger technology, depending on the particular facts and circumstances, without regard to the form of the organization or technology used to effectuate a particular offer or sale. In this Report, the Commission considers the particular facts and circumstances of the offer and sale of DAO Tokens to demonstrate the application of existing U.S. federal securities laws to this new paradigm.

A. Section 5 of the Securities Act

The registration provisions of the Securities Act contemplate that the offer or sale of securities to the public must be accompanied by the "full and fair disclosure" afforded by registration with the Commission and delivery of a statutory prospectus containing information necessary to enable prospective purchasers to make an informed investment decision. Registration entails disclosure of detailed "information about the issuer's financial condition, the identity and background of management, and the price and amount of securities to be offered ? ." SEC v. Cavanagh, 1 F. Supp. 2d 337, 360 (S.D.N.Y. 1998), aff'd, 155 F.3d 129 (2d Cir. 1998). "The registration statement is designed to assure public access to material facts bearing on the value of publicly traded securities and is central to the Act's comprehensive scheme for protecting public investors." SEC v. Aaron, 605 F.2d 612, 618 (2d Cir. 1979) (citing SEC v. Ralston Purina Co., 346 U.S. 119, 124 (1953)), vacated on other grounds, 446 U.S. 680 (1980). Section 5(a) of the Securities Act provides that, unless a registration statement is in effect as to a security, it is unlawful for any person, directly or indirectly, to engage in the offer or sale of securities in interstate commerce. Section 5(c) of the Securities Act provides a similar prohibition against offers to sell, or offers to buy, unless a registration statement has been filed. Thus, both Sections 5(a) and 5(c) of the Securities Act prohibit the unregistered offer or sale of securities in interstate commerce. 15 U.S.C. § 77e(a) and (c). Violations of Section 5 do not require scienter. SEC v. Universal Major Indus. Corp., 546 F.2d 1044, 1047 (2d Cir. 1976).

34 Id.

B. DAO Tokens Are Securities

1. Foundational Principles of the Securities Laws Apply to Virtual Organizations or Capital Raising Entities Making Use of Distributed Ledger Technology

Under Section 2(a)(1) of the Securities Act and Section 3(a)(10) of the Exchange Act, a security includes "an investment contract." See 15 U.S.C. §§ 77b-77c. An investment contract is an investment of money in a common enterprise with a reasonable expectation of profits to be derived from the entrepreneurial or managerial efforts of others. See SEC v. Edwards, 540 U.S. 389, 393 (2004); SEC v. W.J. Howey Co., 328 U.S. 293, 301 (1946); see also United Housing Found., Inc. v. Forman, 421 U.S. 837, 852-53 (1975) (The "touchstone" of an investment contract "is the presence of an investment in a common venture premised on a reasonable expectation of profits to be derived from the entrepreneurial or managerial efforts of others."). This definition embodies a "flexible rather than a static principle, one that is capable of adaptation to meet the countless and variable schemes devised by those who seek the use of the money of others on the promise of profits." Howey, 328 U.S. at 299 (emphasis added). The test "permits the fulfillment of the statutory purpose of compelling full and fair disclosure relative to the issuance of ?the many types of instruments that in our commercial world fall within the ordinary concept of a security.'" Id. In analyzing whether something is a security, "form should