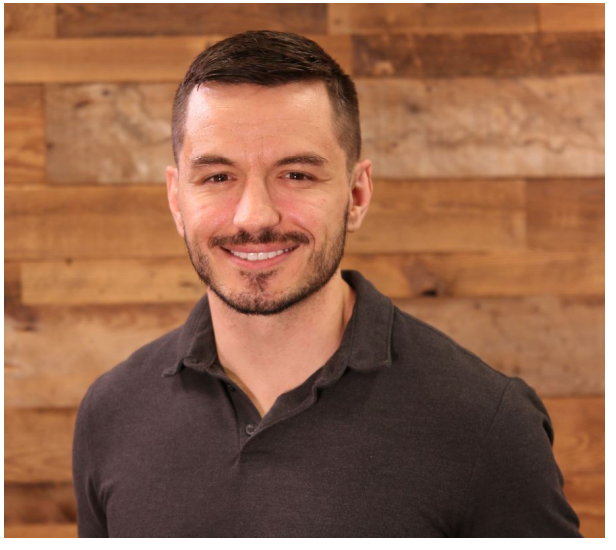
Building Web-Apps with R and SQL Server

R isn't well known for enterprise Web Applications. However, recent advancements in the R ecosystem aim to change that. In this presentation, I will introduce what you need to know for building a production-ready web-app in R.

2019-09-12

About Me

- Matt Sharkey, MBA, MS-BIA, MCSE
- DBA @ BuilderTrend
- Omaha, NE



Data Science Delivery

Challenges

- Users not satisfied with static reports/dashboards, want to interact
- Users want to modify data used in analysis, perform complex what-if analysis
- 200 GB tables as data source
- Need to share results
- A web-based application is an appropriate solution but..
- Team competencies are specific to database and data-analysis, R

Web Development Simplified

- Shiny -> R package for building web apps

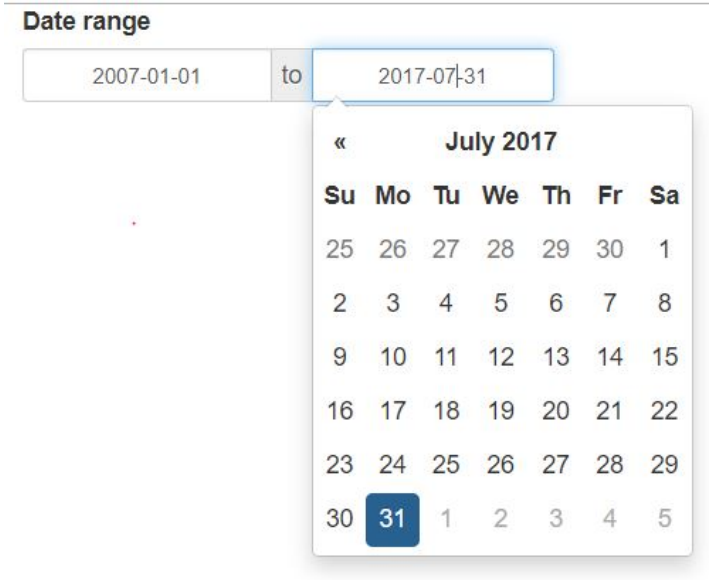
```
<h2>Hello World!</h2>
```

Hello World!

```
library(shiny)

h2('Hello World!')
```

Hello World!

```
dateRangeInput("date"
    ,strong("Date range")
    ,start = "2007-01-01"
    ,end = "2017-07-31"
    ,min = "2007-01-01"
    ,max = "2017-07-31")
```



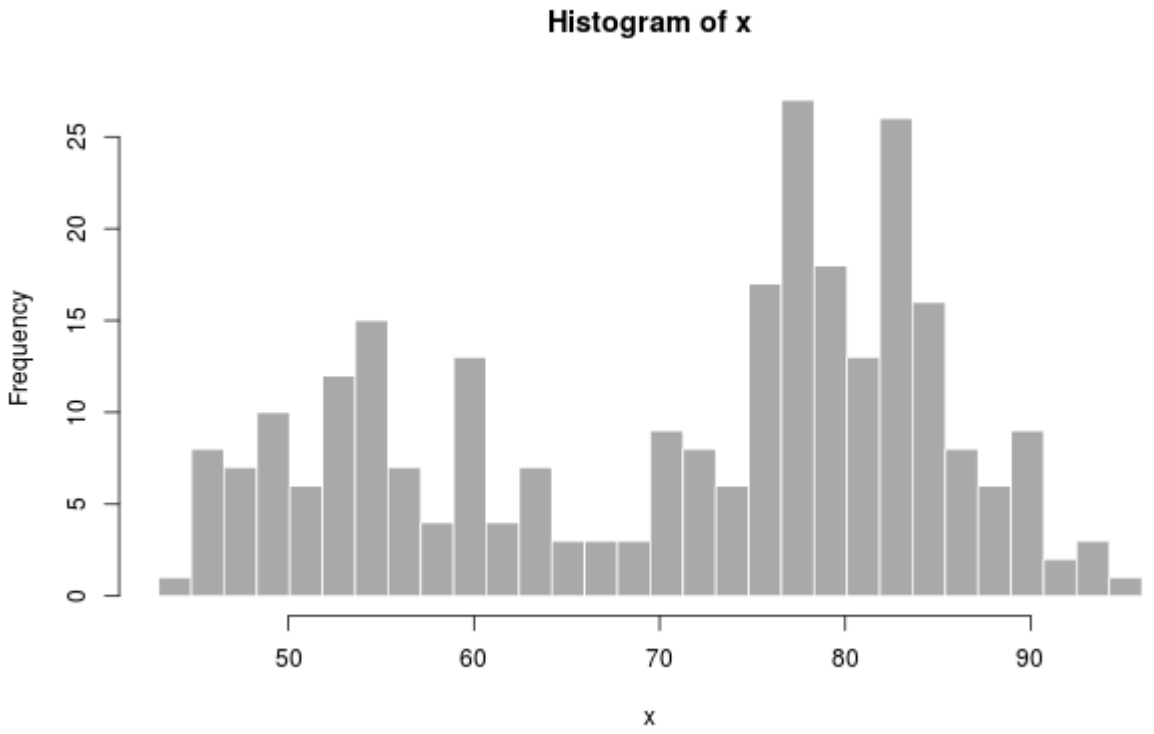

```
<div id="date">
<label class="control-label" for="date"> <strong>Date range</strong>
</label><div class="input-daterange input-group">
<input class="input-sm form-control" type="text"
data-date-week-start="0" data-date-format="yyyy-mm-dd"
data-date-start-view="month" data-min-date="2007-01-01"
data-max-date="2017-07-31"data-initial-date="2007-01-01"
data-date-autoclose="true"/><span class="input-group-addon">
to </span> <input  type="text" data-date-language="en"
data-date-week-start="0" data-date-format="yyyy-mm-dd"
data-min-date="2007-01-01" data-max-date="2017-07-31"/>
  </div></div>
```

Old Faithful Geyser Data

Number of bins:

1 30 50

1 6 11 16 21 26 31 36 41 46 50



More Examples

Shiny Gallery

Production Concerns

- "R is not real programming."

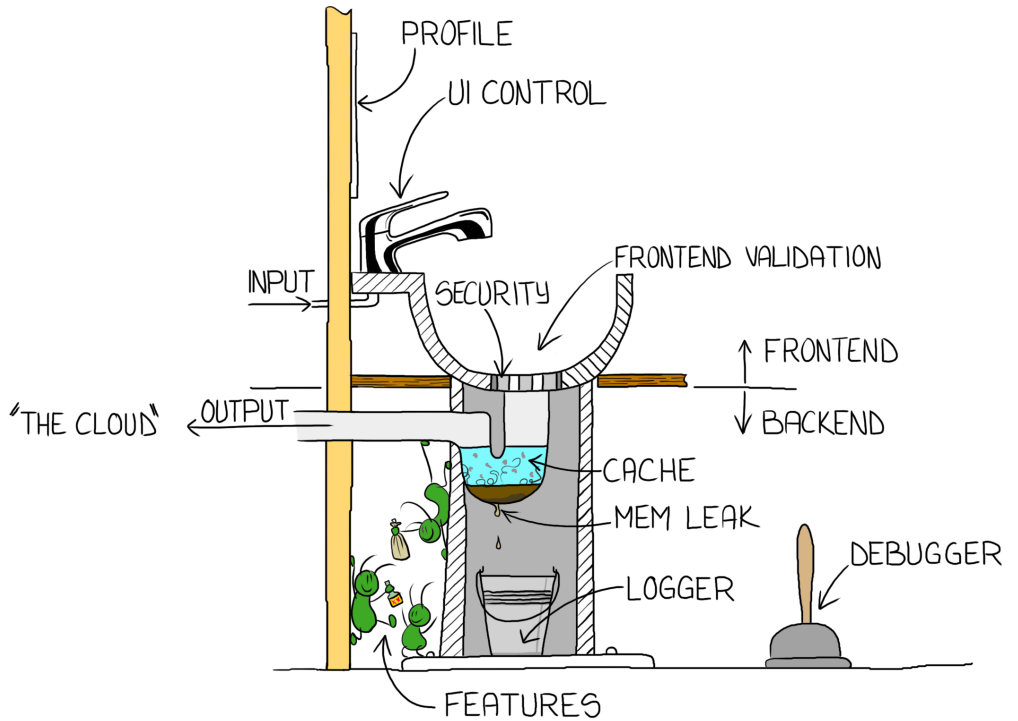
via GIPHY

- "Abstraction is great until something stops working."

Agenda

- Architecture of a Shiny App
- R Primer
- Write Sample App
  - Security
  - Connection Management
  - Profiling
  - Load Testing
  - Caching

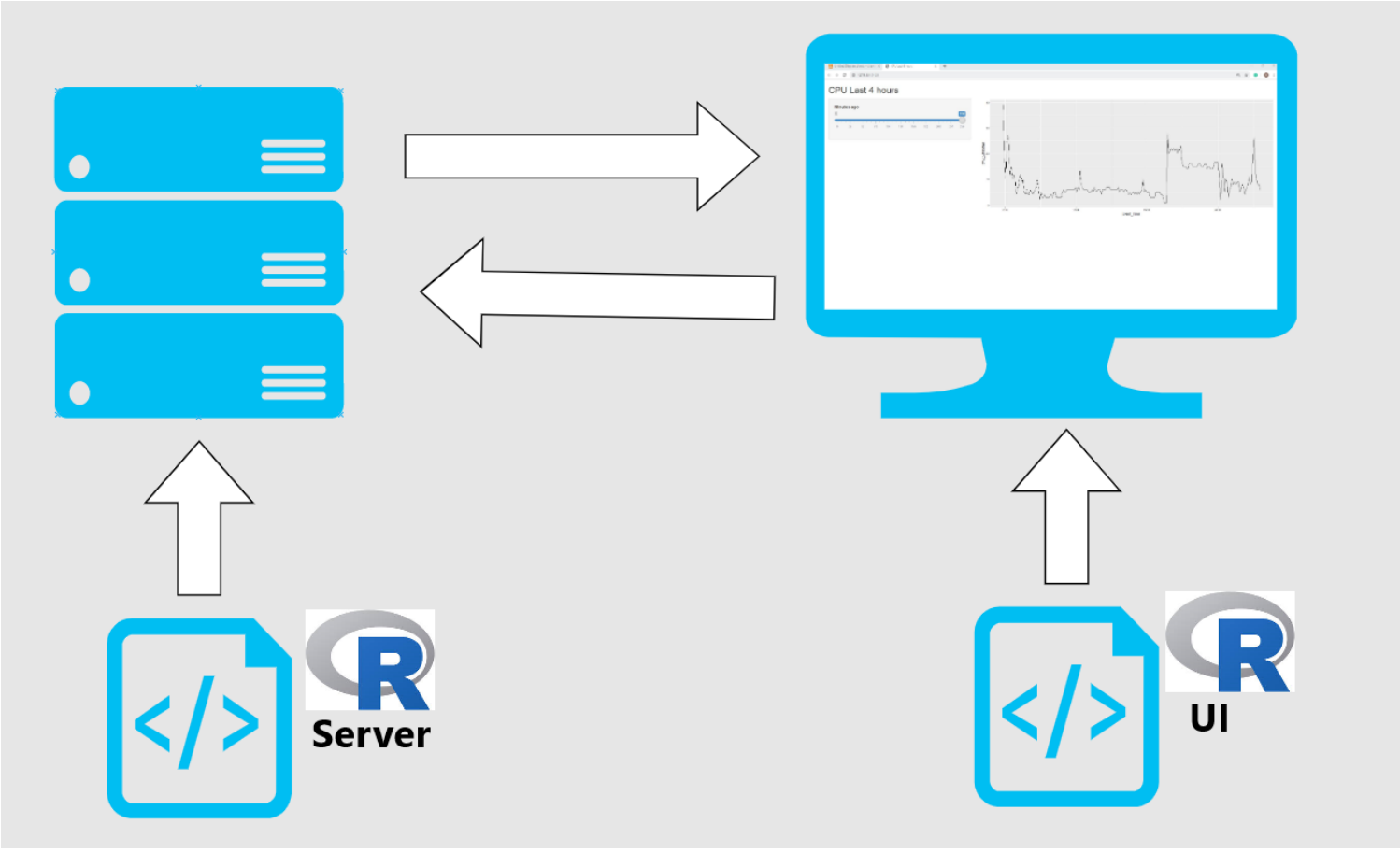
Throwing the Kitchen Sink

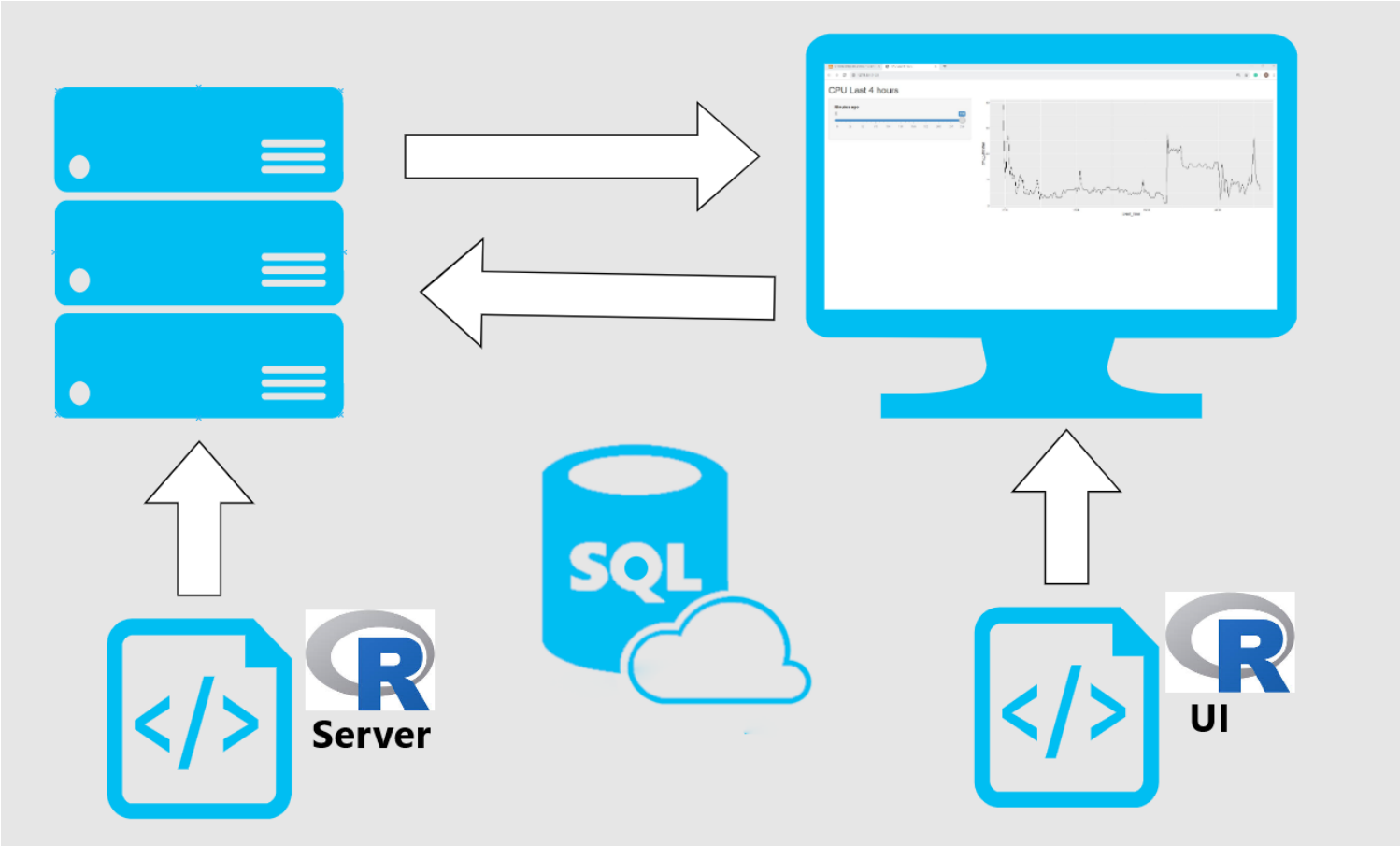


Architecture Model


Server


UI


CPU Last 4 hours
Server
UI


CPU Last 4 hours
SQL
Server
UI

R Language Primer

- Everything is Data or a Function

```
x <- 1

1+x
```

```
## [1] 2
```

```
addFunction <- function(x,x1){
  x+x1
}

addFunction(x,1)
```

```
## [1] 2
```

- Reference functions outside of base R with packages

```
library(DBI)
```

- Checks for a folder name "DBI" in library location

```
.libPaths()[1]
```

```
## [1] "C:/Users/mshar/OneDrive/Old/Documents/R/win-library/3.6"
```

```
help(DBI)
```

DBI

The DBI package defines a common interface between the R and database management systems (DBMS). The interface defines a small set of classes and methods similar in spirit to Perl's DBI, Java's JDBC, Python's DB-API, and Microsoft's ODBC. It defines a set of classes and methods defines what operations are possible and how they are performed:

- connect/disconnect to the DBMS
- create and execute statements in the DBMS
- extract results/output from statements
- error/exception handling
- information (meta-data) from database objects
- transaction management (optional)

Shiny Template

```
library(shiny)
library(DBI)
library(odbc)

con <- dbConnect(drv = odbc(),
                 Driver = 'Sql Server',
                 Server = '.\',
                 Database = 'Test')
```

Shiny Template

```
library(shiny)
library(DBI)
library(odbc)

con <- dbConnect(drv = odbc(),
                 Driver = 'Sql Server',
                 Server = '.\\oldsnapper',
                 Database = 'Test')

my_ui <- fluidPage()
```

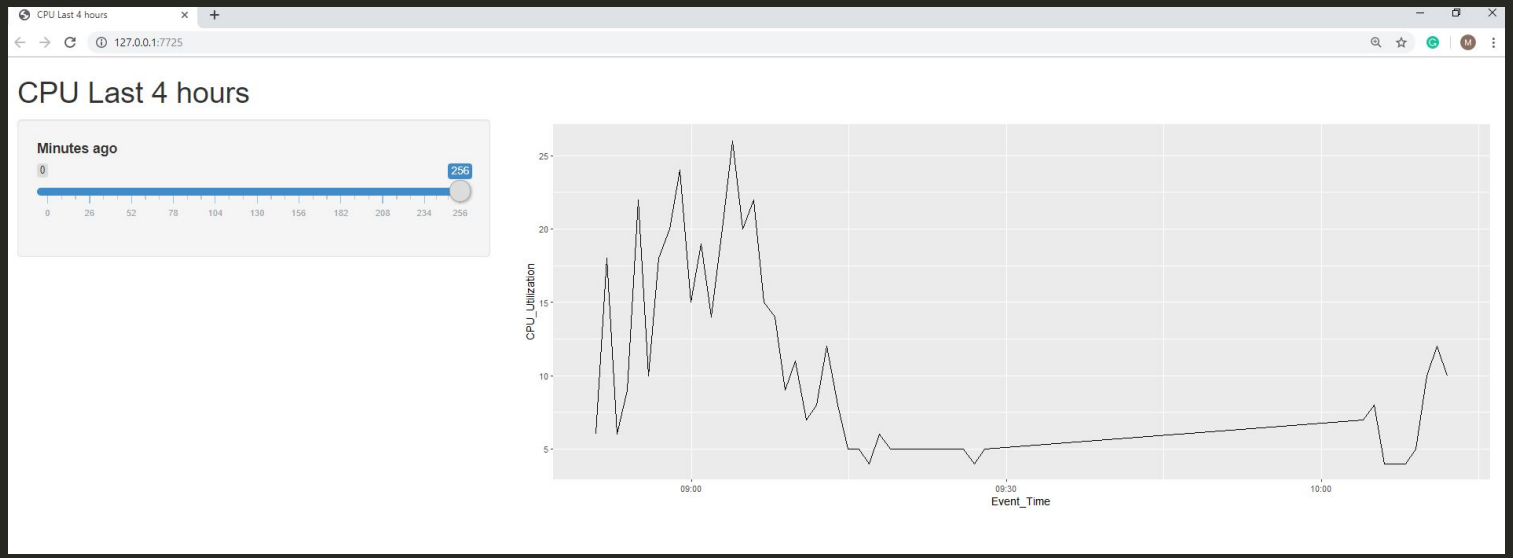
Shiny Template

```
library(shiny)
library(DBI)
library(odbc)

con <- dbConnect(drv = odbc(),
                 Driver = 'Sql Server',
                 Server = '.\\oldsnapper',
                 Database = 'Test')

my_ui <- fluidPage()

my_server <- function(input, output) {}
```

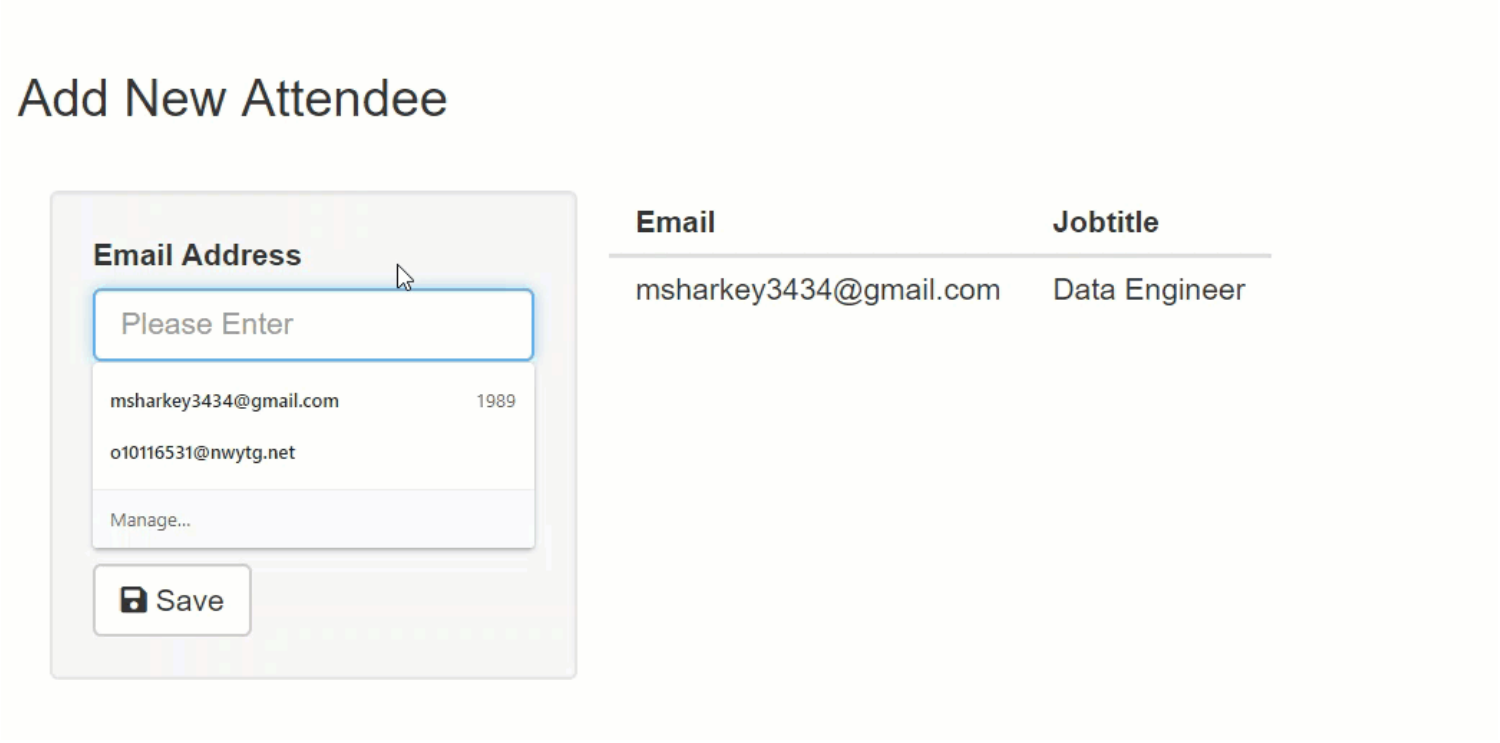
Shiny Template

```
library(shiny)
library(DBI)
library(odbc)

con <- dbConnect(drv = odbc(),
                 Driver = 'Sql Server',
                 Server = '.\\oldsnapper',
                 Database = 'Test')

my_ui <- fluidPage()

my_server <- function(input, output) {}

shinyApp(ui = my_ui, server = my_server)
```

01 - Sample App Demo



Injection Risk



Parameterized

```
  myquery <- paste0("Execute dbo.GetCPUutilization ",input$cpuInput)
```

```
  myquery <- "Execute dbo.GetCPUutilization ?cpu_slider_param"
```

```
myquery <- sqlInterpolate(con,myquery,
           .dots =c(cpu_slider_param <- input$cpu_slider))
```

```
mydata <- dbGetQuery(con,myquery)
```

Building and Managing Connections

Building and Managing Connections

```
con <-  dbConnect(drv = odbc::odbc(),
                  Driver = 'Sql Server',
                  Server = '.\\oldsnapper',
                  Database = 'Test',
                  Trusted_Connection='yes')
    myquery <- ...
    mydata <- dbGetQuery(con,myquery)

     dbDisconnect(con)
```

```
pool <- dbPool(drv = odbc::odbc(),
               Driver = 'Sql Server',
               Server = '.\\oldsnapper',
               Database = 'Test')
my_ui <- ....
my_server <- function(input, output) {
  output$cpuPlot <- renderPlot({
    myquery <- ....
    myqueryint <- ...
     mydata <- dbGetQuery(pool,myqueryint)
```

02 - Building and Managing Connections Demo

Load Testing

App is fast with one user session

What about 50 concurrent users?

```
shinyloadtest::record_session('http://127.0.0.1:6696/')
```

```
java -jar shinycannon-1.0.0-c9c02cb.jar recording.log http://127.0.0.
```

```
df <- shinyloadtest::load_runs("50workers" = "./test-log.997Z")

shinyloadtest::shinyloadtest_report(df, "run1.html")
```

0:00 / 1:51

03 - Review Load Test Results

04 - Caching Demo

Session Recap

- Built a sample app from the template

```
con <- dbConnect(drv = odbc(),
                 Driver = 'Sql Server',
                 Server = '.\\oldsnapper',
                 Database = 'Test')

my_ui <- fluidPage()
my_server <- function(input, output) {}
shinyApp(ui = my_ui, server = my_server)
```

- Profiled an App Session
- Connection Pooling
- Ran a Load Test
- Implemented Caching

Additional Resources

Presentation Repo https://github.com/msharkey/shinybuilder/tree/master/01-SQL_SAT_Presentation

Pluralsight https://app.pluralsight.com/library/courses/r-programming-fundamentals/table-of-contents

Shiny Gallery https://shiny.rstudio.com/gallery/

My E-book https://bookdown.org/msharkey3434/ShinyDB_Book/

My Blog https://www.hinttank.com/

Advanced R book https://amzn.to/32qbna4

People to follow @jcheng,@xieyihui,@hadleywickham,@drob

Thank You

via GIPHY