custom_functions_TAG_ow

```
options(dplyr.summarise.inform = FALSE)

i<-0
install_if_needed <- function(package_name){
  i <<- i+1
  message("Package ", i, " van ", length(required_packages))
  if(!requireNamespace(package_name, quietly = T)){
    install.packages(package_name, quiet = T, dependencies = T, type = "win.binary")
  }
  suppressWarnings(
    suppressMessages(
      library(package_name, character.only = T)
      )
    )

}

required_packages <- c("tidyverse", "RSQLite", "readxl", "openxlsx", "shiny", "shinythemes", "shinycssloaders", "DT", "leaflet")
lapply(required_packages, install_if_needed)

# Functions for loading data files ----------------------------------------
deliverPathToLatestFile <- function(path, file_name_pattern, rank = 1) {
  return(list.files(path, pattern = file_name_pattern, full.names = TRUE) %>%
           as_tibble() %>% arrange() %>% tail(n=rank) %>% head(n=1) %>% as.character)
}

deliverPathToLatestFolder <- function(path, folder_name_pattern, rank = 1) {
  return(dir(path, pattern = folder_name_pattern, full.names = TRUE) %>%
           as_tibble() %>% arrange() %>% tail(n=rank) %>% head(n=1) %>% as.character)
}

# Hard-coded data----------------------------------------------------------
pathToDeelcodes <- file.path('G:', 'OGA', 'G-PROG-ADVIEZEN', 'BEG', 'Team Data', 'apps', 'TAGeR', 'data', 'deelcodes_names.csv')
folderDB <- file.path('G:', 'OGA', 'G-PROG-ADVIEZEN', 'BEG', 'Team Data', 'data', 'db')
pathToDB <- head(sort(list.files(folderDB, pattern = "tag.db$", full.names = T), decreasing = T), 1)
pathToOWDB <- head(sort(list.files(folderDB, pattern = "owt", full.names = T), decreasing = T), 1)
folderOWDB <- file.path('G:', 'OGA', 'G-PROG-ADVIEZEN', 'BEG', 'Team Data', 'data', 'db')
pathToBSQDB <- file.path('G:', 'OGA', 'G-PROG-ADVIEZEN', 'BEG', 'Team Data', 'data', 'db', 'bsqs.db')
folderBSQDB <- file.path('G:', 'OGA', 'G-PROG-ADVIEZEN', 'BEG', 'Team Data', 'data', 'db')

pathToWOZ <- r'{G:\OGA\G-PROG-ADVIEZEN\BEG\Team Data\data\WOZ\output}'

DT_options <- list(autoWidth=TRUE,
                   pageLength = 15,
                   dom = 'ftpr',
                   filter = 'top',
                   ordering = TRUE,
                   language = list(search = 'Zoeken:',
                                   processing = 'Bezig met zoeken...',
                                   paginate = list(previous = 'Vorige',
                                                   `next` = 'Volgende')))

#-----------------------------------------------------------------------------
# Set locations
#-----------------------------------------------------------------------------
pathToRecentOnbezWaardeFile <- deliverPathToLatestFile(pathToWOZ, "_megafile_onbezwaarde_waardetool.*csv")
pathToWOZKadaster <- deliverPathToLatestFile(pathToWOZ, "_WOZkadaster.*csv")
pathToHermesData <- deliverPathToLatestFolder(r'{G:\OGA\G-PROG-ADVIEZEN\BEG\Team Data\data\WOZxBSQ_Hermes\weekup\Output\}','Output_data_weekup')
if(!file.exists(file.path(pathToHermesData,"Aantallen Tabellenupdate met WOZ.csv"))) {
  # next file of weekup after HermesKadaster.csv should exist, otherwise use previous folder
  pathToHermesData <- deliverPathToLatestFolder(r'{G:\OGA\G-PROG-ADVIEZEN\BEG\Team Data\data\WOZxBSQ_Hermes\weekup\Output\}','Output_data_weekup', rank = 2)
}
pathTooutputCasesHernummerd <- file.path(pathToHermesData, "Output Cases Hernummerd.csv")
pathToHermesKadaster <- file.path(pathToHermesData, "HermesKadaster.csv")
pathToHermesData <- deliverPathToLatestFolder(r'{G:\OGA\G-PROG-ADVIEZEN\BEG\Team Data\data\WOZxBSQ_Hermes\weekup\Output\}','Output_data_weekup')
if(!file.exists(file.path(pathToHermesData,"Aantallen Tabellenupdate met WOZ.csv"))) {
  # next file of weekup after HermesKadaster.csv should exist, otherwise use previous folder
  pathToHermesData <- deliverPathToLatestFolder(r'{G:\OGA\G-PROG-ADVIEZEN\BEG\Team Data\data\WOZxBSQ_Hermes\weekup\Output\}','Output_data_weekup', rank = 2)
}


filename_latest_WOZ_Kadaster <- deliverPathToLatestFile(pathToWOZ, '_WOZkadaster.*csv')
filename_latest_hermes <- file.path(pathToHermesData, 'Hermes.csv')
```

```
filename_latest_foutcodes <- file.path(pathToHermesData, 'Foutcodes.csv')
filename_latest_hermes_kadaster <- file.path(pathToHermesData, 'HermesKadaster.csv')
filename_latest_hermes_bestemmingen <- file.path(pathToHermesData, 'Bestemming.csv')
filename_latest_hermes_adressen <- file.path(pathToHermesData, 'Adressen.csv')

#-------------------------------------------------------------------------------
# Load data for OW- & Zoektool
#-------------------------------------------------------------------------------
message("Bestand 1 van de 2")
if(!exists("namesDeelcodes")) {
  namesDeelcodes <- read_delim(pathToDeelcodes,
                               col_types = cols_only(deelcode = col_integer(),
                                                     omschrijving = col_character())
                               , lazy = FALSE)
}

message("Bestand 2 van de 2")
if(!exists("stratenEnBuurten")) {
  stratenEnBuurten <- read_delim(file.path(pathToWOZ, "BAG_TAG_buurtstraten_actueel_historisch.csv"), delim = ';',
                                 col_types = cols_only(GeoCode = col_character(),
                                                       GeoCodeNieuw = col_character(),
                                                       `Naam openbare ruimte` = col_character()
                                 )) %>% pivot_longer(cols = c(GeoCode, GeoCodeNieuw)) %>%
    transmute(Straatnaam = `Naam openbare ruimte`,
              Buurten = str_sub(value, 6, 9),
              Geocode = value,
              oldNew = ifelse(grepl("Nieuw", name), "New", "Old")) %>%
    group_by(Straatnaam) %>%
    distinct(Buurten, Geocode, oldNew) %>%
    ungroup()


  uniqueStreets <- stratenEnBuurten %>%
    select(Straatnaam) %>%
    distinct() %>%
    arrange(Straatnaam)
}



# Custom functions-------------------------------------------------------------
deliverDatedFoldersTibble <- function(aDirList, aRegEx, aDate = "29991231", compareDate = FALSE) {
  aDirList %>%
    as_tibble() %>%
    #Extract date from filename, compute time diff
    transmute(filepath = value,
              dateStr = str_replace_all(filepath, aRegEx, ""),
              fileDate = as.Date(dateStr, "%Y%m%d"),
              compareDate = case_when(compareDate == TRUE ~ as.Date(aDate, "%Y%m%d"),
                                      compareDate == FALSE ~ Sys.Date()),
              timeDifference = difftime(compareDate, fileDate)) %>%
    filter(timeDifference >= 0) %>%
    # break ties with file name
    arrange(timeDifference, desc(filepath))
}

buildObjectQuery <- function(anAddress, aPeiljaar) {
  if(aPeiljaar < 2021) {
  paste0("SELECT ObjectId, GeoCode FROM woz WHERE Adres = '", str_replace(anAddress,"'","''"), "'")
  } else {paste0("SELECT ObjectId, GeoCodeNieuw FROM woz WHERE Adres = '", str_replace(anAddress,"'","''"), "'")}
}

getVersion <- function(pathToDB){
  connection <- dbConnect(RSQLite::SQLite(), pathToDB)
  versionQuery <- "SELECT my_date FROM logs ORDER BY log_id DESC LIMIT 1;"
  if(developerMode){print(versionQuery)}
  version <- format(as.Date(dbGetQuery(connection, versionQuery) %>% pull()), "%d-%m-%Y")
  dbDisconnect(connection)

  version
}

WaarderingsTypeSwitch <- function(aFlag) {
  waarderingsType <- switch (aFlag,
    "WOZ" = "WOZ-waarde",
    "OW" = "Onbezwaarde waarde")

    waarderingsType
}

deliverWaarderingsColumn <- function(aFlag, thisYear) {
  waarderingsColumn <- WaarderingsTypeSwitch(aFlag)

  waarderingsColumn
```

```
}

formatAsDataTable <- function(aTibble, aCaption, aBackgroundColor) {

  aTibble %>%
    datatable(caption = aCaption,
              options = list(scrollX = FALSE,
                             dom = 't',
                             ordering = FALSE),
              class = c('cell-border', 'compact'),
              colnames = c('', ''),
              rownames = FALSE) %>%
    formatStyle(1, backgroundColor = '#F5F5F5')

}

deliverPeiljaar <- function(aPeiljaar, aFlag) {
# 2022-11-24 removed aPeiljaar == "2015" ~ "2014", for WOZ
# so the WOZ-values for peiljaar 2015 can also be found

    if (aFlag == "WOZ") {
      aPeiljaar <- substring(aPeiljaar, 1, 4)

  } else if(aFlag == "BSQ") {
      # the BSQ year in the BSQ database is not the actual BSQ year -2/-3
      aPeiljaar <- case_when(substring(aPeiljaar, 1, 4) == "2015" ~ "2014",
                             aPeiljaar == "2016 overstap (BSQ: 2018)" ~ "2014",
                             aPeiljaar == "2017 overstap (BSQ: 2019)" ~ "2014",
                             aPeiljaar == "2016 niet-overstap (BSQ: 2018)" ~ "2016",
                             aPeiljaar == "2017 niet-overstap (BSQ: 2019)" ~ "2017",
                             TRUE ~ substring(aPeiljaar, 1, 4))
  }  else if(aFlag == "OW") {
       aPeiljaar <- case_when(substring(aPeiljaar, 1, 4) == "2015" ~ "2014",
                              TRUE ~ substring(aPeiljaar, 1, 4))
  }
  aPeiljaar
}

writeWOZNotification <- function(aPeiljaar) {

  previousYear <- as.character(-1 + as.numeric(format(Sys.Date(), "%Y")))
  nog <- ifelse(substring(aPeiljaar, 1, 4) == previousYear, "nog", "")
  ifelse(substring(aPeiljaar, 1, 4) %in% c(previousYear, "2020"), sprintf(wozjaarNotification, aPeiljaar, nog), "")

}

writeNevenadresNotification <- function(aStraat, aHuisnummer, aToevoeging) {
  adres <- trimws(paste(aStraat, aHuisnummer, aToevoeging))
  nevenadressen <- getNevenadressen(adres, "Amsterdam")

  if(length(nevenadressen)>1){
    paste('<p style="color:#DC143C;font-family:architect;text-align:left;font-size:8">
<br>Hoofdadres: ', nevenadressen[1], '<br>Nevenadres(sen):', paste(nevenadressen[-1], collapse = ', '), '</li><br>Mogelijk zijn de WOZ-waarden te vinden bij h
  }else{""}
}

getNevenadressen <- function(adres, woonplaats, include_hoofdadres = TRUE) {

  # function that uses PDOK locatieserver-API to find nevenadressen for a specific address/city
  # will return all address(es) connected to this address regardless of the amount of nevenadressen
  # 1st: find the address and the corresponding aoa_id
  # 2nd: find how many addresses are connected to the aoa_id
  # 3rd: return these addresses as list:
  # first address is hoofdadres (if include_hoofdadres = TRUE)

  q_url <- paste0("https://api.pdok.nl/bzk/locatieserver/search/v3_1/free?q=('",adres,"'+and+woonplaatsnaam:",woonplaats,")&fq=type:adres&wt=json")
  q_url <- gsub(" ", "+", q_url)
  if(developerMode) print(q_url)
  json <- jsonlite::read_json(q_url, simplifyVector = TRUE)

  adres_response <- try(strsplit(paste(json$response$docs["weergavenaam"][[1]][1]), ",")[[1]][1])
  if(!("try-error" %in% class(adres_response))) {
    if(grepl(adres, adres_response)==TRUE | grepl(adres, gsub("-", " ", adres_response)) == TRUE){
      aoa_id <- json$response$docs["adresseerbaarobject_id"][[1]][1]
      q_url <- paste0("https://api.pdok.nl/bzk/locatieserver/search/v3_1/free?q=",aoa_id)
      if(developerMode) print(q_url)
      json <- jsonlite::read_json(q_url, simplifyVector = TRUE)

      nevenadressen <- list()
      hoofdadres <- list()
      for(i in 1:length(json$response$docs[[1]])){
        adres <- paste(json$response$docs["straatnaam"][[1]][i], json$response$docs["huis_nlt"][[1]][i])
        if("gekoppeld_perceel" %in% colnames(json$response$docs)){
          if(is.null(json$response$docs["gekoppeld_perceel"][[1]][i][[1]])) {
```

```
            nevenadressen <- append(nevenadressen, adres)
          }else{
            hoofdadres <- append(hoofdadres, adres)
          }
        }else{
          hoofdadres <- append(hoofdadres, adres)
        }
      }
      if(include_hoofdadres){
        return(c(hoofdadres, nevenadressen))
      }else{
        return(nevenadressen)
      }
    }
  }else{
    return(list())
  }
}

deliverBSQ <- function(aBSQ) {
  ifelse(aBSQ == 99999999 | length(aBSQ) == 0,
         '-',
         paste0(aBSQ, "%"))
}

deliverCode <- function(aCode) {
  ifelse(aCode == 99999999 | aCode == 9999,
         '-',
         aCode %>% as.character())
}

deliverWoningDeelcode <- function(aDeelcodesDF, aDeelobject) {
  thisIndex <- match(aDeelobject, aDeelcodesDF[['omschrijving']])
  thisDeelcode <- aDeelcodesDF[thisIndex,][[1]] %>%
    as.character()

  if(thisDeelcode == 9999) {
    thisDeelcode <- '-'
  }

  thisDeelcode
}

deliverBuurtnaam <- function(anAddressDF, aGeocode) {
  if(!grepl("\\d\\d", str_sub(aGeocode,-2,-1))) {
  thisIndex <- match(aGeocode, anAddressDF[['GeoCode']])
  anAddressDF[thisIndex,][['Naam Buurt']]
  } else if(grepl("\\d\\d", str_sub(aGeocode,-2,-1))) {
    thisIndex <- match(aGeocode, anAddressDF[['GeoCodeNieuw']])
    anAddressDF[thisIndex,][['Naam buurt nieuw']]
  }
}

deliverOWDBTable <- function(aGebruikscode) {

  ifelse(aGebruikscode == "STWO", "stwo", "egwo")
}

deliverOWPerM2 <- function(aStreet, aBuurtcode, aDeelcode, aGebruikscode, aPeiljaar, aPathToDB) {

  # Get street code first
  con <- dbConnect(RSQLite::SQLite(), aPathToDB)
  streetCodeQuery <- paste0("SELECT Straatcode FROM straten WHERE Straatnaam = '", str_replace(aStreet,"'","''"), "'")

  if(developerMode){print(streetCodeQuery)}
  streetCode <- dbGetQuery(con, streetCodeQuery)

  # Now get onbezwaarde waarde per m2
  hulpCode <- paste0(streetCode, aBuurtcode, aDeelcode)

  thisTable <- deliverOWDBTable(aGebruikscode)

  owQuery <- paste0("SELECT Waarde FROM ", thisTable,
                    " WHERE Hulp = '", hulpCode,
                    "' AND WOZjaar = ", aPeiljaar)

  if(developerMode){print(owQuery)}
  onbezwaarde <- dbGetQuery(con, owQuery)

  # Close DB connection
  dbDisconnect(con)

  onbezwaarde$Waarde
}
```

```r
addUnits <- function(aNumber, aDeelcode) {

  thisUnit <- ifelse(aDeelcode %in% c('1350', '1490', '1500', '1590', '1600', '1840', '1860', '1900', '3340'),
                " eenh.",
                " m\u00B2")

  aNumber %>%
    as.character() %>%
    paste0(thisUnit)
}


deliverCorrectieFactor <- function(aPeiljaar, aGebruikscode, aSpecialWoning, aDeelcode, aSurface) {

  discountFactor <- deliverDiscountFactor(aSpecialWoning)

  if(aDeelcode == 1100 & aSurface == 25) {
    correctie <- 1
  } else if(aDeelcode == 1100 & aGebruikscode == "STWO") {
    correctie <- 1.7652*aSurface^(-0.176)
  } else if(aDeelcode == 1100 & aGebruikscode == "EGWO") {
    correctie <- 2.0246*aSurface^(-0.218)
  } else {correctie <- 1}

  # voor peiljaar 2018 geldt een correctiefactor van 0.98
  if(aPeiljaar == '2018') {
    correctie <- correctie*0.98
  }

  correctie * discountFactor
}


deliverDiscountFactor <- function(aSpecialWoning) {
  switch(aSpecialWoning,
         'Studentenwoning' = 0.85,
         'Bejaardenwoning' = 0.85,
         'Containerwoning' = 0.6,
         1)
}

roundDown <- function(aNumber, aPeiljaar) {
  aUnit <- ifelse(aPeiljaar %in% c('2014', '2015', '2016'), 500, 1000)
  aNumber - (aNumber %% aUnit)
}


# Onbezwaarde Waarde functions-------------------------------------------------------------
deliverOnbezwaardeWaardeObjectData <- function(aGebruikscode, aStreet, aBuurtcode,
                                               aSpecialWoning, aPeiljaar,
                                               aDeelObject1, aSurfaceOrQuantity1,
                                               aDeelObject2, aSurfaceOrQuantity2,
                                               aDeelObject3, aSurfaceOrQuantity3,
                                               aDeelObject4, aSurfaceOrQuantity4,
                                               aDeelObject5, aSurfaceOrQuantity5,
                                               aDeelObject6, aSurfaceOrQuantity6,
                                               aDeelObject7, aSurfaceOrQuantity7,
                                               aPathToDB, aPathToBSQDB, aPathToOWDB) {




  # Return Peiljaar string
  aPeiljaarBSQ <- deliverPeiljaar(aPeiljaar, "BSQ")

  aPeiljaarWOZ <- deliverPeiljaar(aPeiljaar, "OW")

  con <- dbConnect(RSQLite::SQLite(), aPathToOWDB)
  peiljaarQuery <- "SELECT WOZjaar FROM egwo ORDER BY Hulp LIMIT 100"

  if(developerMode){print(peiljaarQuery)}
  peiljaarOW <- dbGetQuery(con, peiljaarQuery) %>% distinct()
  oldDBNewPeiljaarFlag <- (!(aPeiljaarWOZ %in% (peiljaarOW$WOZjaar)) & !(aPeiljaarWOZ < 2021))
  dbDisconnect(con)

  if(oldDBNewPeiljaarFlag) {
    thisNotification <- oldDBNewPeiljaarNotification
  }

  theseKenmerkValues <- c(aGebruikscode, rep(NA, 3))

  if(!(aStreet=="")) {
    theseKenmerkValues[[2]] <- aStreet
```

```
}

if(!(aBuurtcode=="")) {
  theseKenmerkValues[[3]] <- aBuurtcode
}

if(!(aSpecialWoning=="Nee: regulier")) {
  theseKenmerkValues[[4]] <- aSpecialWoning
}

kenmerken <- tibble(label = c('Gebruikscode', 'Straat', 'Buurtcode', 'Bijzonderheid'),
                    value = theseKenmerkValues)

kenmerkenDataTable <- kenmerken %>%
  na.omit() %>%
  formatAsDataTable(tags$caption('Kenmerken woning', style = "color:#2F4F4F"), '#F5F5F5')


if(nrow(kenmerken) > 2) {

  con <- dbConnect(RSQLite::SQLite(), aPathToOWDB)

  # Get Straatcode from owt.db
  straatcodeQuery <- paste0("SELECT Straatcode FROM straten WHERE Straatnaam = '",
                            str_replace(aStreet,"'","''"), "'")

  if(developerMode){print(straatcodeQuery)}
  straatCode <- dbGetQuery(con, straatcodeQuery)

  dbDisconnect(con)

  # Get version
  OWVersion <- getVersion(aPathToOWDB)
  thisVersion <- paste0('<p style="color:#2F4F4F;font-family:architect;text-align:right;font-size:8"> <b>Versie: ', OWVersion, '</b></p>')

  geocode <- paste0(straatCode, aBuurtcode)
  # Get Buurtnaam from woz db
  conWOZ <- dbConnect(RSQLite::SQLite(), aPathToDB)
  # Get wijkgegevens uit addressDict
  if(as.numeric(aPeiljaarWOZ) < 2021) {
    BuurtWijkQuery <- paste0("SELECT * FROM woz WHERE GeoCode = '",geocode, "'")
    if(developerMode){print(BuurtWijkQuery)}
    BuurtWijkData <- dbGetQuery(conWOZ, BuurtWijkQuery)
    buurtNaam  <- BuurtWijkData %>% distinct(`Naam Buurt`) %>% pull()
    wijkGegevens <- BuurtWijkData %>% transmute(`Code Wijk` = str_sub(geocode, 6,8), `Naam Wijk`) %>% distinct()
  } else {
    BuurtWijkQuery <- paste0("SELECT * FROM woz WHERE GeoCodeNieuw = '", geocode, "'")
    if(developerMode){print(BuurtWijkQuery)}
    BuurtWijkData <- dbGetQuery(conWOZ, BuurtWijkQuery)
    buurtNaam  <- BuurtWijkData %>% distinct(`Naam buurt nieuw`) %>% pull()
    wijkGegevens <- BuurtWijkData %>% transmute(`Code Wijk nieuw` = str_sub(geocode, 6,7), `Naam Wijk nieuw`) %>% distinct()
  }

  # Get BSQ from tag.db
  bsqQuery <- paste0("SELECT Waarde, Soort FROM bsqs WHERE GeoCode = '",
                     geocode, "' AND Type = '", aGebruikscode, "' AND",
                     " WOZjaar = ", aPeiljaarBSQ
  )

  conBSQ <- dbConnect(RSQLite::SQLite(), aPathToBSQDB)

  if(developerMode){print(bsqQuery)}
  bsqTibble <- dbGetQuery(conBSQ, bsqQuery)
  bsqValue <- bsqTibble %>% select(Waarde)

  if(!oldDBNewPeiljaarFlag) {
    if(nrow(bsqValue) == 0) {
      thisBSQ <- "-"
      thisNotification <- bsqNotification
    } else {
      if(is.na(bsqValue)) {
        thisBSQ <- "-"
        thisNotification <- bsqNotification
      } else {
        thisBSQ <- paste0(bsqValue, '%')
        if(grepl("Maatwerk", bsqTibble$Soort)) {
          thisNotification <- str_replace(maatwerkNotification, "%1", paste(unique(bsqTibble$Soort), collapse = ", "))
        } else {
          thisNotification <- ''
        }
      }
    }
  } else {thisBSQ <- "-"}
  dbDisconnect(conBSQ)
```

```
    if(identical(buurtNaam, character(0))) buurtNaam <- ""
    #Tibble met buurtGegevens
    buurtGegevens <- tibble(label = c('Straatcode', 'Buurtstraat', 'Buurt', 'BSQ'),
                            value = c(straatCode, geocode, buurtNaam, thisBSQ))

    buurtenDataTable <- buurtGegevens %>%
      formatAsDataTable(tags$caption('Buurtgegevens', style = "color:#2F4F4F"), '#F5F5F5')

    wijkDataTable <- wijkGegevens %>%
      formatAsDataTable(tags$caption('Wijkgegevens', style = "color:#2F4F4F"), '#F5F5F5')

    waarderingDataTable <- buildWaarderingData(aStreet, aBuurtcode,
                                               aGebruikscode, aSpecialWoning,
                                               aDeelObject1, aSurfaceOrQuantity1,
                                               aDeelObject2, aSurfaceOrQuantity2,
                                               aDeelObject3, aSurfaceOrQuantity3,
                                               aDeelObject4, aSurfaceOrQuantity4,
                                               aDeelObject5, aSurfaceOrQuantity5,
                                               aDeelObject6, aSurfaceOrQuantity6,
                                               aDeelObject7, aSurfaceOrQuantity7,
                                               aPeiljaarWOZ, aPathToOWDB, OWVersion)



    return(list(kenmerkenDataTable,
                buurtenDataTable,
                waarderingDataTable,
                thisNotification,
                writeWOZNotification(aPeiljaar),
                wijkDataTable
           ))

  }

  return(NA)
}

buildWaarderingData <- function(aStreet, aBuurtcode,
                                aGebruikscode, aSpecialWoning,
                                aDeelObject1, aSurfaceOrQuantity1,
                                aDeelObject2, aSurfaceOrQuantity2,
                                aDeelObject3, aSurfaceOrQuantity3,
                                aDeelObject4, aSurfaceOrQuantity4,
                                aDeelObject5, aSurfaceOrQuantity5,
                                aDeelObject6, aSurfaceOrQuantity6,
                                aDeelObject7, aSurfaceOrQuantity7,
                                aPeiljaar, aPathToOWDB, aVersionDate) {


  # Initialise waardering table and vectors
  colNames <- setNames(rep("", 6), c('Deelcode', 'Deelobject', 'Oppervlakte/Aantal', 'Waarde', 'Correctiefactor', 'Onbezwaarde waarde'))
  waardering <- bind_rows(colNames)[0, ]

  deelcodesSet <- rep(NA, 8)
  deelobjectenSet <- rep(NA, 8)
  surfaceOrQuantitiesSet <- rep(NA, 8)
  owPerSurfaceSet <- rep(NA, 8)
  correctieFactorenSet <- rep(NA, 8)
  onbezwaardeWaardeSet <- rep(NA, 8)
  onbezwaardeWaardes <- rep(NA, 8)


  if(!(aDeelObject1 == "")) {

    deelcode1 <- deliverWoningDeelcode(namesDeelcodes, aDeelObject1)
    owPerSurface1 <- deliverOWPerM2(aStreet, aBuurtcode, deelcode1, aGebruikscode, aPeiljaar, aPathToOWDB)
    if(length(owPerSurface1) == 0) {owPerSurface1 <- '-'}
    correctieFactor1 <- deliverCorrectieFactor(aPeiljaar,aGebruikscode, aSpecialWoning, deelcode1, aSurfaceOrQuantity1)
    onbezwaardeWaardes[[1]] <- correctieFactor1 * aSurfaceOrQuantity1 * suppressWarnings(as.numeric(owPerSurface1[[1]]))


    deelcodesSet[[1]] <- deelcode1
    deelobjectenSet[[1]] <- aDeelObject1
    surfaceOrQuantitiesSet[[1]] <- addUnits(aSurfaceOrQuantity1, deelcode1)
    owPerSurfaceSet[[1]] <- owPerSurface1 %>% as.character()
    correctieFactorenSet[[1]] <- correctieFactor1 %>% round(digits = 6) %>% as.character() %>% str_replace("[.]", ",")
    onbezwaardeWaardeSet[[1]] <- onbezwaardeWaardes[[1]] %>% round(digits = 0) %>% as.character()
  }
  if(!(aDeelObject2 == "")) {

    deelcode2 <- deliverWoningDeelcode(namesDeelcodes, aDeelObject2)
    owPerSurface2 <- deliverOWPerM2(aStreet, aBuurtcode, deelcode2, aGebruikscode, aPeiljaar, aPathToOWDB)
    if(length(owPerSurface2) == 0) {owPerSurface2 <- '-'}
```

```
  correctieFactor2 <- deliverCorrectieFactor(aPeiljaar,aGebruikscode, aSpecialWoning, deelcode2, aSurfaceOrQuantity2)
  onbezwaardeWaardes[[2]] <- correctieFactor2 * aSurfaceOrQuantity2 * suppressWarnings(as.numeric(owPerSurface2[[1]]))

  deelcodesSet[[2]] <- deelcode2
  deelobjectenSet[[2]] <- aDeelObject2
  surfaceOrQuantitiesSet[[2]] <- addUnits(aSurfaceOrQuantity2, deelcode2)
  owPerSurfaceSet[[2]] <- owPerSurface2 %>% as.character()
  correctieFactorenSet[[2]] <- correctieFactor2 %>% round(digits = 6) %>% as.character() %>% str_replace("[.]", ",")
  onbezwaardeWaardeSet[[2]] <- onbezwaardeWaardes[[2]] %>% round(digits = 0) %>% as.character()
}

if(!(aDeelObject3 == "")) {

  deelcode3 <- deliverWoningDeelcode(namesDeelcodes, aDeelObject3)
  owPerSurface3 <- deliverOWPerM2(aStreet, aBuurtcode, deelcode3, aGebruikscode, aPeiljaar, aPathToOWDB)
  if(length(owPerSurface3) == 0) {owPerSurface3 <- '-'}
  correctieFactor3 <- deliverCorrectieFactor(aPeiljaar,aGebruikscode, aSpecialWoning, deelcode3, aSurfaceOrQuantity3)
  onbezwaardeWaardes[[3]] <- correctieFactor3 * aSurfaceOrQuantity3 * suppressWarnings(as.numeric(owPerSurface3[[1]]))

  deelcodesSet[[3]] <- deelcode3
  deelobjectenSet[[3]] <- aDeelObject3
  surfaceOrQuantitiesSet[[3]] <- addUnits(aSurfaceOrQuantity3, deelcode3)
  owPerSurfaceSet[[3]] <- owPerSurface3 %>% as.character()
  correctieFactorenSet[[3]] <- correctieFactor3 %>% round(digits = 6) %>% as.character() %>% str_replace("[.]", ",")
  onbezwaardeWaardeSet[[3]] <- onbezwaardeWaardes[[3]] %>% round(digits = 0) %>% as.character()
}

if(!(aDeelObject4 == "")) {

  deelcode4 <- deliverWoningDeelcode(namesDeelcodes, aDeelObject4)
  owPerSurface4 <- deliverOWPerM2(aStreet, aBuurtcode, deelcode4, aGebruikscode, aPeiljaar, aPathToOWDB)
  if(length(owPerSurface4) == 0) {owPerSurface4 <- '-'}
  correctieFactor4 <- deliverCorrectieFactor(aPeiljaar,aGebruikscode, aSpecialWoning, deelcode4, aSurfaceOrQuantity4)
  onbezwaardeWaardes[[4]] <- correctieFactor4 * aSurfaceOrQuantity4 * suppressWarnings(as.numeric(owPerSurface4[[1]]))

  deelcodesSet[[4]] <- deelcode4
  deelobjectenSet[[4]] <- aDeelObject4
  surfaceOrQuantitiesSet[[4]] <- addUnits(aSurfaceOrQuantity4, deelcode4)
  owPerSurfaceSet[[4]] <- owPerSurface4 #%>% as.character()
  correctieFactorenSet[[4]] <- correctieFactor4 %>% round(digits = 6) %>% as.character() %>% str_replace("[.]", ",")
  onbezwaardeWaardeSet[[4]] <- onbezwaardeWaardes[[4]] %>% round(digits = 0) %>% as.character()
}

if(!(aDeelObject5 == "")) {

  deelcode5 <- deliverWoningDeelcode(namesDeelcodes, aDeelObject5)
  owPerSurface5 <- deliverOWPerM2(aStreet, aBuurtcode, deelcode5, aGebruikscode, aPeiljaar, aPathToOWDB)
  if(length(owPerSurface5) == 0) {owPerSurface5 <- '-'}
  correctieFactor5 <- deliverCorrectieFactor(aPeiljaar,aGebruikscode, aSpecialWoning, deelcode5, aSurfaceOrQuantity5)
  onbezwaardeWaardes[[5]] <- correctieFactor5 * aSurfaceOrQuantity5 * suppressWarnings(as.numeric(owPerSurface5[[1]]))

  deelcodesSet[[5]] <- deelcode5
  deelobjectenSet[[5]] <- aDeelObject5
  surfaceOrQuantitiesSet[[5]] <- addUnits(aSurfaceOrQuantity5, deelcode5)
  owPerSurfaceSet[[5]] <- owPerSurface5# %>% as.character()
  correctieFactorenSet[[5]] <- correctieFactor5 %>% round(digits = 6) %>% as.character() %>% str_replace("[.]", ",")
  onbezwaardeWaardeSet[[5]] <- onbezwaardeWaardes[[5]] %>% round(digits = 0) %>% as.character()
}

if(!(aDeelObject6 == "")) {

  deelcode6 <- deliverWoningDeelcode(namesDeelcodes, aDeelObject6)
  owPerSurface6 <- deliverOWPerM2(aStreet, aBuurtcode, deelcode6, aGebruikscode, aPeiljaar, aPathToOWDB)
  if(length(owPerSurface6) == 0) {owPerSurface6 <- '-'}
  correctieFactor6 <- deliverCorrectieFactor(aPeiljaar,aGebruikscode, aSpecialWoning, deelcode6, aSurfaceOrQuantity6)
  onbezwaardeWaardes[[6]] <- correctieFactor6 * aSurfaceOrQuantity6 * suppressWarnings(as.numeric(owPerSurface6[[1]]))

  deelcodesSet[[6]] <- deelcode6
  deelobjectenSet[[6]] <- aDeelObject6
  surfaceOrQuantitiesSet[[6]] <- addUnits(aSurfaceOrQuantity6, deelcode6)
  owPerSurfaceSet[[6]] <- owPerSurface6 #%>% as.character()
  correctieFactorenSet[[6]] <- correctieFactor6 %>% round(digits = 6) %>% as.character() %>% str_replace("[.]", ",")
  onbezwaardeWaardeSet[[6]] <- onbezwaardeWaardes[[6]] %>% round(digits = 0) %>% as.character()
}

if(!(aDeelObject7 == "")) {

  deelcode7 <- deliverWoningDeelcode(namesDeelcodes, aDeelObject7)
  owPerSurface7 <- deliverOWPerM2(aStreet, aBuurtcode, deelcode7, aGebruikscode, aPeiljaar, aPathToOWDB)
  if(length(owPerSurface7) == 0) {owPerSurface7 <- '-'}
  correctieFactor7 <- deliverCorrectieFactor(aPeiljaar,aGebruikscode, aSpecialWoning, deelcode7, aSurfaceOrQuantity7)
  onbezwaardeWaardes[[7]] <- correctieFactor7 * aSurfaceOrQuantity7 * suppressWarnings(as.numeric(owPerSurface7[[1]]))

  deelcodesSet[[7]] <- deelcode7
```

```
    deelobjectenSet[[7]] <- aDeelObject7
    surfaceOrQuantitiesSet[[7]] <- addUnits(aSurfaceOrQuantity7, deelcode7)
    owPerSurfaceSet[[7]] <- owPerSurface7 %>% as.character()
    correctieFactorenSet[[7]] <- correctieFactor7 %>% round(digits = 6) %>% as.character() %>% str_replace("[.]", ",")
    onbezwaardeWaardeSet[[7]] <- onbezwaardeWaardes[[7]] %>% round(digits = 0) %>% as.character()
  }

  # 2023-11-14 round down to 500s/1000s
  if(!(length(na.omit(onbezwaardeWaardes)) == 0)) {

    deelcodesSet[[8]] <- ''
    deelobjectenSet[[8]] <- ''
    surfaceOrQuantitiesSet[[8]] <- ''
    owPerSurfaceSet[[8]] <- ''
    correctieFactorenSet[[8]] <-'Totaal'
    onbezwaardeWaardeSet[[8]] <- sum(onbezwaardeWaardes, na.rm = TRUE) %>% roundDown(aPeiljaar) %>% as.character()

  }



  waarderingDataTable <- waardering  %>%
    add_row(Deelcode = deelcodesSet,
            Deelobject = deelobjectenSet,
            `Oppervlakte/Aantal` = surfaceOrQuantitiesSet,
            Waarde = owPerSurfaceSet,
            `Correctiefactor` = correctieFactorenSet,
            `Onbezwaarde waarde` = onbezwaardeWaardeSet) %>%
    na.omit() %>%
    datatable(caption = tags$caption(paste0('Waardering met peiljaar ', aPeiljaar, ' [Versie: ', aVersionDate, ']'), style = "color:#2F4F4F"),
              options = list(scrollX = FALSE,
                             dom = 't',
                             ordering = FALSE,
                             initComplete = JS(jscHeader),
                             language = list(zeroRecords = 'Geen gegevens beschikbaar'),
                             columnDefs = list(list(className = 'dt-center', targets = "_all"))),
              class = c('stripe', 'cell-border'),
              rownames = FALSE) %>%
    formatCurrency(c('Waarde', 'Onbezwaarde waarde'),
                   currency = '\U20AC',
                   mark = '.',
                   digits = 0) %>%
    formatStyle('Correctiefactor',
                target = 'row',
                backgroundColor = styleEqual(c('Totaal'), c('#FAEBD7')))

}



# javascript functions-------------------------------------------------------

jscHeader <- "function(settings, json) {
$(this.api().table().header()).css({'background-color': '#2F4F4F', 'color': '#F8F8FF'});
}"

# HTML#----------------------------------------------------------------------
bsqNotification <-
  '<p style="color:#DC143C;font-family:architect;text-align:left;font-size:8"> &#9888; Let op: BSQ is onbekend</p>'
maatwerkNotification <-
  '<p style="color:#DC143C;font-family:architect;text-align:left;font-size:8"> &#9888; Let op: Dit is een maatwerk-BSQ (%1)!</p>'
logoTAGeR <-
  '<p style="color:#2F4F4F;font-family:garamond;padding-left:70px;"> <b>TAG&#8495;&#8477; </b></p>'
#  'TAG&#8495;&#8477;'
homeGreeting <-
  '<br><br><br><br><br>
  <h1 style="color:#2F4F4F;font-family:garamond;padding-left:10px;"> <b>TAG&#8495;&#8477; </b></h1><br>
  <h3 style="color:#2F4F4F;font-family:garamond;font-size:14"> Welkom en veel plezier!</h3>'
```