**Video Script: Section 8 Video 5 – building our dashboard.**

|  |  |  |  |
| --- | --- | --- | --- |
| No. | Description | Action on screen | Narration |
| 1 | Introduction  (Outcome and why it is desirable)  This should give the viewer an idea of the outcome of the task at the beginning of the videos and set the stage and expectations of the viewer. | Refer to PPT | In this video, we’re putting everything together and build our dashboard. |
| 2 | Context(Problem/Solution)  Present the viewer with a real-world solution and how the situation would pose as a challenge. It always helps to draw the viewer's attention using a use-case. Metadata template can be used here. |  | We have made a Shiny app with the economic indicator’s time series, another shiny app with the bubble chart and know how to change the user interface given which tab is active. We’re ready to build our dashboard. |
| 3 | Guidance (How to do it and how it works): | Launch R and open dashboard/UI.R, dashboard/server.R and dashboard/global.R in the editor. | Launch R and open dashboard/UI.R, dashboard/server.R and dashboard/global.R in the editor. |
| 4 |  | Switch to global.R | Global.R contains the lookup tables for translating the indicator code into more descriptive names.  As both UI.R and server.R need those, this is the best place to put them.  A third vector translates the code to a fuller textual description. |
| 5 |  | Switch to UI.R | UI.R has the now familiar structure: header, sidebarPanel and mainPanel. |
| 6 |  | Highlight where necessary. | The sidebarPanel has two conditional panels that appear/disappear depending on which tab is active. |
| 7 |  |  | mainPanel has a tabset with two tabPanels:  a plot (timeSeries / bubbleChart) and  a verbatim text (descriptionTab1 / descriptionTab2)  Thanks to the tabSet id (id=’theTabs’), we know which tab is active.  The tab are identified with their parameter ‘value’ and the conditions are written in very simple javascript.  input.theTabs == 'firstTab'  and  input.theTabs == 'secondTab'  A dot is used instead of a $ for accessing input$theTabs. |
| 8 |  | Switch to server.R | For efficiency’s sake, a number of things are set outside shinyServer(). All this code is only run once, when the server is launched.   * the data is loaded. * The function bubbleChart is defined. This is the code we built in the third video. * The function printDescription is defined.   Both functions are used in shinyServer() but are not reactive so they’re better defined outside shinyServer().  In shinyserver(), we define the 4 necessary outputs:   * The rendering of the bubblechart and the time series, with some extra code for customizing the plots. * The rendering of the text boxes with the full descriptions of one or two indicators. |
| 9 |  | Run in the command line:  shiny::runApp(“dashboard”)  A description... | Our dashboard is ready!  You can launch it and start exploring this real dataset from the world bank with:  shiny::runApp(“dashboard”) |
| 10 |  |  |  |
| 11 |  |  |  |
| 12 |  |  |  |
| 13 |  |  |  |
| 14 |  |  |  |
| 15 |  |  |  |
| 16 | Conclusion:The video concludes by showing the viewer that the goal has been achieved, and reminding them why they should be happy about that. A PowerPoint summary slide with the key points emphasized would make it easier for the viewer to remember what was covered in the video | Back to PPT | Using what we’ve learned in this course, from ggplot2 to shiny, we built an interactive dashboard for exploring some economic indicators.  You are now equipped to build graphs and interactive webpages in R for your own data! |