— P&O CW 2023–2024: Story Book Report —

Department of Computer Science – KU Leuven **October 15, 2024**

Team <00>: <git-tag-link>

<first name last name>	0h
<first name last name>	0h
...	

Fill in the estimated time spent on P&O CW per team member for the full semester up till now, including the lab sessions.

Focus points

This part will be filled in by the teaching team.

	0	1	2	3
Structure & organization				
	0	1	2	3
Writing style				
	0	1	2	3
Technical depth				

Practical guidelines

- This report should be no longer than 4 pages.
- Add your team number, names and time spent at the top of the report. Remove the angular brackets in the heading.
- Mark the author of a section or a paragraph in the margin using `marginpar`. You can abbreviate the names.
- Delete all the blue guidelines and the section “LaTeX Tips & Tricks" from your final report before submitting.

Content guidelines

- The main deliverable of the Story Book assignment is the Reverse Dictionary Search feature: Given a description of a word, inputted via the command line, present a list of words that match this description.
- In this report, we want you to document how your team tackled this problem. Think of it as a miniature of the report you will be submitting at the end of the semester. Pay attention to its structure. Use academic language and writing style.

- Discuss and evaluate your solution on a technical level. What did you implement? How did you implement it, and why? How did you test it? How good does the implementation work? You should of course report on the results as well.
- Also discuss the learning process. We want to know how you approached the task, e.g., how tasks were divided, which difficulties were encountered, ... You can also discuss technical difficulties encountered with Git, Node and/or Typescript.
- Write the report for an audience with similar background knowledge as yourself. Avoid over-explaining basic concepts, but remember that a report should state the problem at hand, such that it is self-contained.

1 LaTeX Tips & Tricks

KB,DD

Mention authors in the margin.
There is a lot of things you can do with LaTeX. We give examples of how to include figures, tables, code snippets, . . . , as well as how to reference them and use citations.

1.1 Tables and referencing

KB

Using the `booktabs` package, you can create professional looking tables. An example is given in Table 1, which we were just able to reference automatically using `\ref{...}`. You can reference many things, as long as you give them a label using `\label{...}`. Even though numbering may change, the reference will always be correct, like the fact that this is Subsection 1.1.

Table cells are separated with `&` and rows are ended with `\\`. The construction looks like this in general:

```
\begin{table}[h]
  \centering % comment: center the table
  \begin{tabular}{...}
    ...
  \end{tabular}
  \caption{...}\label{...}
\end{table}
```

After `\begin{tabular}`, you specify the number of columns and their alignment, e.g. `{l c c c c}`.

	explanation			
Topic	col 1	col 2	col 3	col 4
hor. lines	`\toprule`	`\midrule`	`\bottomrule`	`\cmidrule(l){2-6}`
multicolumn	'explanation'	is done with	`&\multicolumn`	`{4}{c}{...}`
positioning	`[h]` (here)	`[t]` (top)	`[b]` (bottom)	`[p]` (separate page)

Table 1: This is inside a `\caption{...}` construction and following it by `\label{...}` allows you to reference this table.

KB,DD

1.2 Figures and citations

Figures can be included using the `graphicx` package. An example is given in Figure 1. The construction looks like this in general:

```
\begin{figure}[h]
  \centering
  \includegraphics[width=<ratio>\textwidth]{<file>}
  \caption{...}\label{...}
\end{figure}
```

You can also create figures in LaTeX using TikZ [2]. An example is given in Figure 2. In this case, you use `\input{...}` instead of `\includegraphics{...}` to include the TikZ figure defined in a separate file.

Figure 1: Always remember to explain what is in the figure. This is a picture of the Gitlab icon.

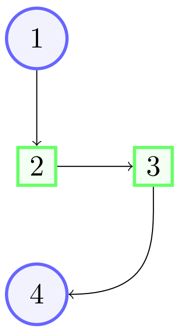

Figure 2: TikZ example.

Notice how we added an automatic citation to the Tikz website in the previous paragraph? This is done using the `\cite{...}` command. The references are managed in a separate file, `references.bib`, and are included at the end of the document using the `\bibliography{...}` command. If you add the proper information in the `.bib` file, the citation will be automatically formatted. The style of the bibliography can be changed by changing the argument of the `\bibliographystyle{...}` command. Remember to put the citations in the right place in the text though, e.g. if I cite the `pgfplots` package [1], I put the citation right after, and not after the whole sentence.

DD

1.3 Equations and code snippets

There are a lot of ways to include equations in LaTeX. You can use the `equation` environment to include a single equation, or the `align` environment to include multiple equations.

An example is given in Equation 1. Alternatively, you can use the `$$...$$` or `\[...\]` constructions. To insert an equation like $E = mc^2$ or $F = ma$ directly in the text, you can use `$...$` or `\( ...\)`.

$$\begin{split}\nabla \cdot \vec{E} &= \frac{\rho}{\varepsilon_0} \\ \nabla \times \vec{E} &= -\frac{\partial \vec{B}}{\partial t}\end{split} \tag{1}$$

To reference an equation like '(`<number>`)', you can use `\eqref{...}`: In (1) we actually used the `split` environment from the `amsmath` package to align the equations. If we had used the `align` environment, we would have gotten a number for each line. We can also remove the numbering by adding an asterisk to the environment name, e.g.

```
\begin{equation*}
  ...
\end{equation*}
```

would not have numbered the equation. In general, you only want to number the equations you are going to reference later.

Throughout the examples, we have listed LaTeX code snippets using the `verbatim` environment. To include code snippets from a programming language, you can use the `listings` package. An example of how to do this is

```
\begin{lstlisting}[ language=ES6,
                    caption={Example of a code snippet.},
                    label={lst:example}]
  const example = (a, b) => {
    return a + b;
  }
\end{lstlisting}
```

which results in Listing 1. The `language` parameter specifies the language of the code, which is used to highlight the code properly. The `caption` and `label` parameters are used to give the code snippet a caption and a label, respectively. We have configured the `ES6` language option in the preamble to highlight JavaScript code.

```
const example = (a, b) => {
return a + b;
}
```

Listing 1: Example of a code snippet.

References

[1] Pgfplots package. https://www.overleaf.com/learn/latex/Pgfplots_package. Accessed: 2021-10-29.

[2] pgf - a portable graphic format for tex. https://github.com/pgf-tikz/pgf. Accessed: 2020-10-26.