IJNRD.ORG ISSN : 2456-4184
INTERNATIONAL JOURNAL OF NOVEL RESEARCH AND DEVELOPMENT (IJNRD) | IJNRD.ORG
An International Open Access, Peer-reviewed, Refereed Journal

A WEB-BASED MEDIUM FOR DYNAMIC AND INTERACTIVE PRESENTATION

[1]G Lavanya, [2]Lakshmi Priya S

[1]Assistant Professor, [2]Assistant Professor
[1]Computer Science and Engineering,
[1]Sathyabama Institute of Science and Technology, Chennai, India

Abstract: Online presentation in a virtual class has two critical needs: the presenter should be able to make changes in the presentation dynamically and the participants should be able to interact with the presentation. Without the interaction of the participants, the presenter has no feedback cues to alter her presentation dynamically to suit the condition of the participants. Existing online presentation tools only allow sharing screen, streaming video, and mixing quizzes. There is still a need for a coherent web-based medium to build dynamic and interactive presentations. Runestone is an ecosystem for interactive textbooks. With interactive books, readers can interact with them. This involves bringing in a HTML5 based presentation system such as reveal.js features into runestone. First, the stylesheets need to be designed for presentation. The stylesheet will have to be designed to make the webpages as a presentation made of a sequence of slides instead of continuous pages. Secondly, the webpages will have to be synchronized such that the presenter will be able synchronize the viewing for all the participants. To mitigate the tedious HTML code of the presentation, the presentation will be prepared in a lighter markup language such as Restructured text (rst) and then, using Python scripts, the rst file will be converted to html file. There will be two modes namely user control mode and presenter control mode. In presenter control mode, the presenter will control the navigation of the slides for all the users. In user control mode, the user is free to navigate to any page of the presentation and interact with it. This project develops a web-based tool with these features: a presentation is a sequence of web pages; the presentation can be prepared using an extensible markdown language; the presenter can synchronize the viewing of the pages or, as the need arises, let the users navigate the pages freely; the presenter can make changes and execute programs in the page, which will be executed in a server; participants can make changes and execute programs in the pages, which will be executed in the browser itself; presentation can have a variety of questions (MCQ, short answer, program snippets) and participants can answer these questions with immediate helpful feedback; the responses of the participants to the questions will give feedback to the presenter as to the effectiveness of the presentation.

Keywords - Restructured text, Reveal JS, Sphinx

With a conventional presentation tool, the presenter prepares the contents ahead of the “talk” and, at the time of meeting in a room, presents it to an audience. At best, the presenter can animate the contents. The presenter cannot make any changes to the content dynamically, at the time of presentation. The audience just listens to the presentation. When translated to an online setting, the audience simply watch the presentation. Online presentation has opened both challenges and opportunities. An immediate challenge to the presenter is that she has no feedback cues from the audience and thus has no way of changing her “tone” to suit the condition of the listeners. There is a compelling need to help both the presenter and the participants. Fortunately, the very online medium that has created the challenge also affords us the opportunity to meet it.

A web-based medium allows the presenter to make changes in the presentation and show the effects of the changes immediately. In addition, the presenter can ask the participants to make changes in the “presentation” and see the effects for themselves. Participants can also be enabled to interact with the presentation in several other ways. A use-case in Computer Science education is teaching a programming course. It is imperative for the presenter to be able to construct or modify programs in front of the audience and run it. It is more compelling to ask the participants to construct and modify programs and run them immediately.

A video, however interesting, falls short of meeting these challenges. A web-based medium – using web technologies such as HTML5, CSS3, JavaScript – is very pliable for incorporating these features and more.

2.1 Usability and Usage of Interactive Features in an online E book for CS Teachers

This paper reports the results from initial studies of on eBook. First, they have conducted a usability study comparing three different eBook platforms. Next, they conducted a study of teacher use of the eBook. Their effort is aimed towards providing high-quality computing education that meets the needs and prior knowledge of potential teachers, in an online setting that can fit into their lives.

They built their eBook using the Runestone Interactive platform. Runestone eBooks are accessed through a Web browser. They have sections and chapters, like a traditional book. Each chapter appeared as a single web page to the participants. While the eBooks can have videos embedded in them, they are not structured around video lectures, in contrast to a lecture-based MOOC. Runestone books contain interactive elements, which allow them to go beyond simple PDF or EPUB eBooks. Users can edit and execute Python programs within the pages of the eBooks.

They were concerned with the legibility and navigation usability, and with the usability of the interactive elements. They conducted a survey of teachers, asking them to judge the eBook on these elements. The four interactive features that they asked their participants to rate were:

- Active Code widget which allows the user to edit and execute Python code, displaying any results or output, in the web browser.
- Code Lens widget which acts as a code visualization tool that allows the user to step through the code. It also displays variable values and program output.
- Parsons Problems widget which allows the user to drag and drop blocks of code, from a bank of code blocks, into the correct order.
- Multiple Choice widget which asks the user a question about a coding concept, code, or output. It provides feedback explaining why the selected answer is correct or incorrect.

From their usability survey, the eBook was found to be legible, easy to navigate, and usable. The interactive features were as good as, if not better than, comparable features in other eBooks. Their usability participants did not find it difficult to use the eBook. However, their usage study teachers (who did have less experience with computing than those who participated in the design study) did occasionally struggle with the interface, such as not attempting to solve any of the Parson problems.

2.2 On the Value of Using an Interactive Electronic Textbook in an Introductory Programming Course

In this paper, the authors report their initial observations on the changes noticed in students' motivational factors and learning results when a static PDF textbook was replaced by an interactive textbook in a large introductory service course. They found increase in both motivational factors, as well as learning gain. In addition, students' feedback on the learning resources improved.

The contents and requirements of the course in two years (2015 and 2016) were the same. The main difference was that in 2015, there were 18 lectures together covering all course topics, and the main learning resource was a static PDF textbook written by the lecturer. In 2016, the number of the lectures was reduced to 10, the lectures concentrated on basic concepts, and the students were asked to self-study the more advanced examples using an interactive textbook which was developed from the previous year static resource. The text in both books was almost the same. However, the e-text book also included interactive animations, and annotated code examples. Moreover, in 2016, there was also a direct link from the exercises to the corresponding textbook chapter while in 2015, no such link existed, and there was only the whole PDF book available as a resource.

However, usage of e-textbooks is a complex phenomenon, and students are not using them in isolation. Their current research is a pilot study concerning the usage of an e-textbook. While they observed several positive results both in motivation and learning results, they should treat these results with care. Only a minority of students have followed the lectures to the end in both years, which implies that this effect might be small. On the other hand, the clear increase of using the e-textbook, covering almost the whole 2016 student cohort, suggests that students considered the resource useful and used it when working with the exercises. For future research they need to log more accurately how students accessed the chapters and how much they used the interactive elements in the book.

2.3 Identifying Design Principles for CS Teacher E books Through Design-Based Research

The interactive elements in the eBook were designed based on research results from educational psychology and are intended to make learning more efficient and effective. They use a design-based research methodology to frame their work. Design-based research is an iterative approach to educational research that grew out of the work of Ann Brown, an educational psychologist, and Alan Collins, a learning scientist. The goal of design-based research is to both improve practice and generate or advance theories that transcend the context.

They tested the usability of the interactive features, active code, code lens, Parson's problems, and multiple-choice questions against those of similar eBooks (Zyante and CS Circles). Most of the teachers in the study preferred the Runestone user interface for all but one of the interactive features (the code lens). The design principles listed here are the ones that they have the most confidence in based on their multiple iterations, teacher observations, teacher feedback, teacher interviews, log file data analyses, and laboratory experiments. Evidence from the eBook interactions and experiments supports their belief that building upon educational psychology design principles is the right first step in developing their eBooks.

The limitation is that they do not have the end of chapter tests for many of the teachers who worked through the eBook, so they cannot comment on how much the teachers learned from using the eBook. In addition, they are basing some of their recommendations on teacher feedback; however, they only received feedback from thirty-eight teachers. The contribution of this paper is a set of design principles that others can use when developing interactive eBooks for computer science teachers. They have presented evidence to support using a worked examples plus practice approach. The eBook provided worked examples using the active code and code lens features, and practice problems using multiple-choice, Parson's problems, and fill-in-the-blank questions.

PROBLEM STATEMENT

To construct a web-based presentation medium with the following capabilities:
1. The presentation can be prepared in an extensible and programmable markdown language which permits extending the markdown as need arises
2. Presenter and participants can view the presentation synchronously
3. Presenter or the participants can make changes in the presentation dynamically, such as changing a program fragment and executing it
4. Participants can interact with the presentation
5. It provides feedback to the presenter about the effectiveness of the presentation
6. Presentation should not consume high bandwidth
7. Presenter's audio should be incorporated to the finished presentation.

SCOPE

Instead of having a one-way presentation, the project aims at developing a two-way live and interactive presentation. "Live" means that code can be executed and "interactive" means student can respond. The scope of the project is to use and/or develop JavaScript libraries for interactive presentation that encourages and motivates students to learn.

OBJECTIVES

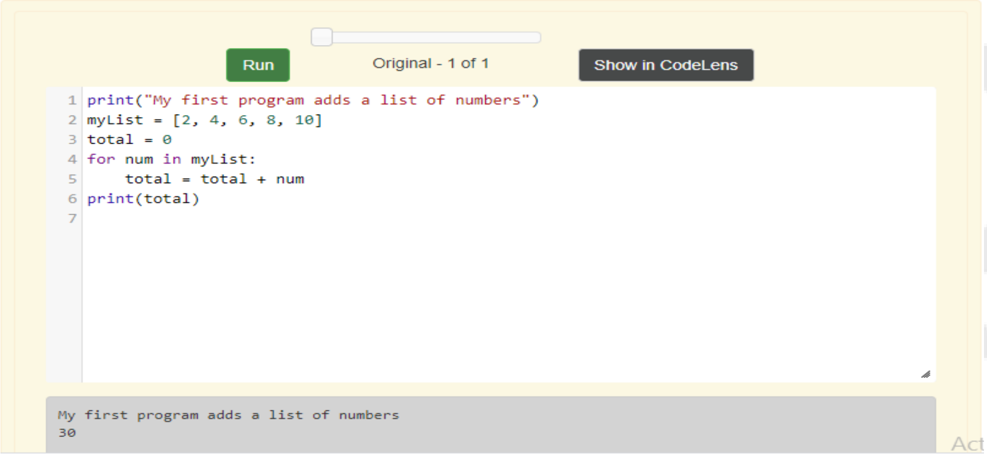
The objective of the project is stated as follows:
1. Instead of using Runestone interactive books we want to add runestone interactive presentation, that means we must bring reveal.js feature into runestone.
2. We use the JavaScript modules of Runestone Interactive and those of reveal.js to enable preparation of interactive presentation.
3. We can author the presentation in Restructured Text (rst) and translate the presentation from rst to HTML.
4. Presenter or the participants can make changes in the presentation dynamically, such as changing a program fragment and executing it
5. Presenter and participants can interact with the presentation.

SYSTEM FEATURES INTERACTIVE WIDGETS

Runestone Interactive is a framework composed of JavaScript library and a web application to support interactive, open-source textbooks for computer science, math, and other topics. It is a simple integrated LMS to give instructors insight into their students' performance. It provides instructors the tools for writing their own interactive activities (or even their own books!). It also supports an application server for hosting their content on the web. We can also use this server to provide textbooks as a service.

Active Code

Typical textbooks allow us to read about programming, but do not allow us to practice. The web-based tool called active code allows us to write, modify, and execute programs right in the text itself (right from the web browser). Although this is certainly not the way real programs are written, it provides an excellent environment for learning a programming language like Python since we can experiment with the language as we are reading, as shown in the screenshot of Figure 3.1.



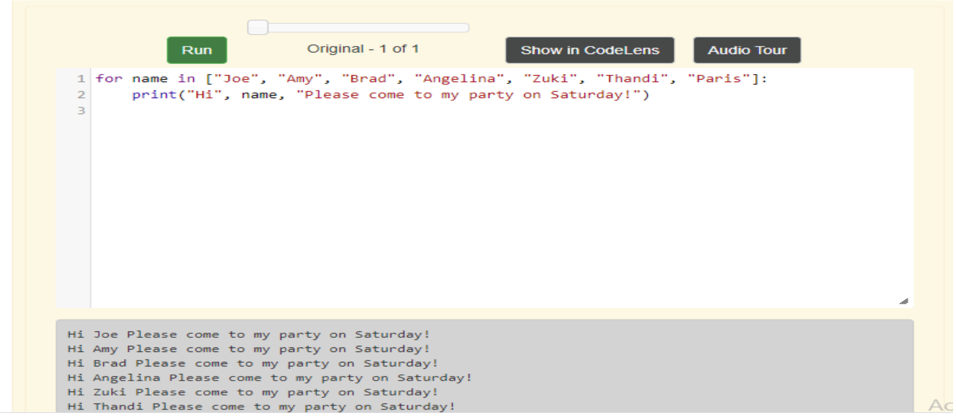
```
print("My first program adds a list of numbers")
myList = [2, 4, 6, 8, 10]
total = 0
for num in myList:
    total = total + num
print(total)
```

```
My first program adds a list of numbers
30
```

Figure 3.1 Active Code Example

Audio Tours

Active code things may also embrace audio tours. These highlight one or a lot of lines of code as they play audio describing that code.



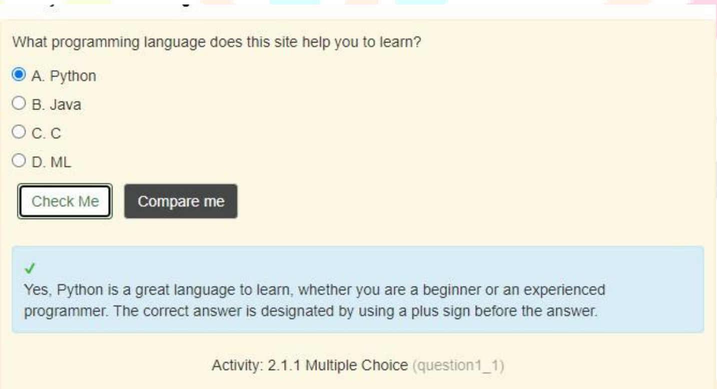
```
for name in ["Joe", "Amy", "Brad", "Angelina", "Zuki", "Thandi", "Paris"]:
    print("Hi", name, "Please come to my party on Saturday!")

```

```
Hi Joe Please come to my party on Saturday!
Hi Amy Please come to my party on Saturday!
Hi Brad Please come to my party on Saturday!
Hi Angelina Please come to my party on Saturday!
Hi Zuki Please come to my party on Saturday!
Hi Thandi Please come to my party on Saturday!
```

Figure 3.2 Audio tours

MULTIPLE CHOICE QUESTIONS

It is also possible to embed simple questions into the text. These questions provide a way for the students to check themselves as they go along. The questions also provide feedback so that we can understand why an answer may or may not be correct.


What programming language does this site help you to learn?

● A. Python
○ B. Java
○ C. C
○ D. ML

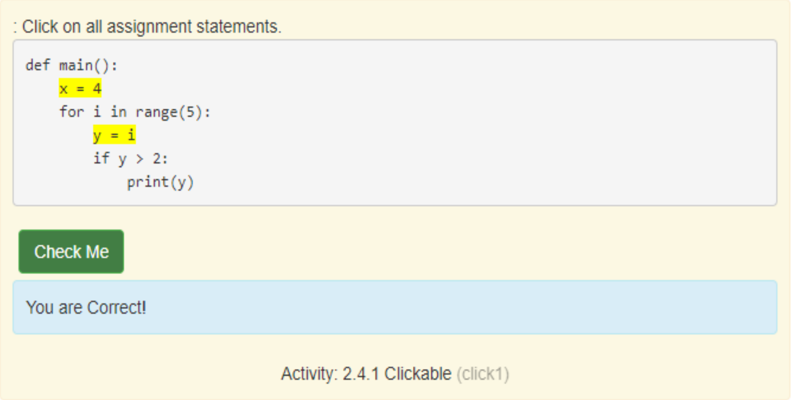
✓
Yes, Python is a great language to learn, whether you are a beginner or an experienced programmer. The correct answer is designated by using a plus sign before the answer.

Activity: 2.1.1 Multiple Choice (question1_1)

Figure 3.4 MCQ

CLICKABLE AREA

Clickable area is another type of assess question we can use. All one needs to do is to write the code we wish to be in the question and wrapthe parts that we would like to be clickable in either a click-correct tag or a click-incorrect tag. Students can click on all elements we defined as being clickable and will be evaluated once they click the 'Check Me' button.


: Click on all assignment statements.

```
def main():
    x = 4
    for i in range(5):
        y = i
        if y > 2:
            print(y)
```

You are Correct!

Activity: 2.4.1 Clickable (click1)

Figure 3.5 Clickable area

METHODOLOGY

One of the existing systems is Runestone server for interactive text- books. Interactive book means the reader will be able interact with the con- tents of the book, i.e., while reading the book, there can be questions andthe reader must respond to them. There can be questions such as multiple choice and short answers. After answering the question, the review of answer is given in the book, so that you can check your answers. While read- Ing books, readers will answer the question given in the textbook and then Runestone will collect the answer given by readers. In this approach, the instructor can know about how much of the book the reader has studied and progressed. An interactive book, apart from the questions and answers, can also have programming questions. Instead of reading the book, the book can have a snippet of code that the reader can simply run or change the code and run. Instead of the book form, we intend to create the contents in the presentation form so that the instructor can synchronize the slides with the readers and the readers can interact with the presentation.

In proposed system, Runestone will support interactive book. However, we need to add interactive presentation using Runestone. First, we will need to develop JavaScript modules to change web pages from continuous pagesto a sequence of slides. Second, the instructor should be able to synchronize viewing of the slides of all the participants. These are mainly useful for online education.

Writing the contents of the presentation in HTML is tedious with its tags. Therefore, we have decided to author the presentation in a light markup language such as Restructured Text and convert it to HTML document by writing a translator in Python.

It is mainly used both online and offline. We will have two modes namely user control mode and presenter control mode. In presenter control mode, the presenter will control the navigation of the slides for all the users. In user control mode, the user is free to navigate to any page of the presentation and interact with it.

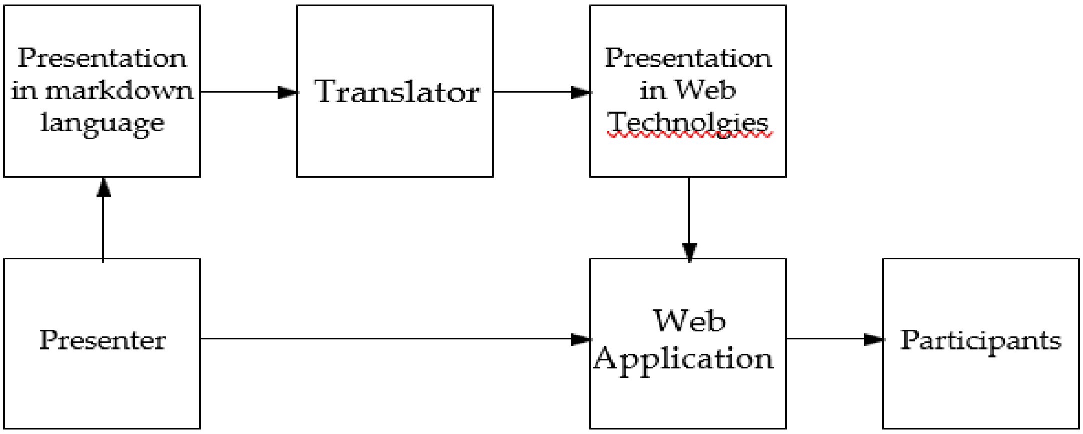

Presentation in markdown language
Translator
Presentation in Web Technolgies
Presenter
Web Application
Participants

Figure 4.1 Block diagram of the web-based presentation system

The overall system is outlined in Figure 4.1 and the main componentsare as follows:

- We need a language for expressing the presentation. We can use any one of the existing mark-down languages, such as markdown, Restructured Text, org mode. We have decided to use Restructured Text (rst) since we have Python modules to translate the markdown from rst to HTML. However, a Domain Specific Language (DSL) designed as a mark-down language designed specifically for presentation will be a cleaner solution.

- We will use HTML5 and the related technologies such as CSS3 and JavaScript as the target language for the presentation. There are several HTML5 solutions for presentation such as Reveal.js, Remark (using React framework), and Eagle.js (using Vue.js framework). A HTML5 and JS framework must be put together for making the presentationas a web application.

REVEAL.JS

Reveal.js-plugins is an open-source HTML presentation framework. It isa tool that enables anyone with a web browser to create fully featured and beautiful presentations for free. This collections includes the following plug- ins: Anything: A plugin for adding plots, charts, animated SVGs,or anything else inside an HTML object using a JSON string and a JavaScript function. Audio slideshow: A plugin for audio playback and recording. Chalkboard: A plugin adding a flat solid and slide annotation. shows created with reveal.js ar designed on open net technologies. meaning something you will do on the web; you'll neutralize your presentation. modification designs with CSS, embody Associate in Nursing external web content victimization Associate in Nursing iframe or add your own custom behavior victimization our JavaScript API. The framework comes with a broad vary of options together with nested slides, Markdown support, Auto-Animate, PDF export, speaker notes, LaTeX support, and syntax highlighted code.

RESTRUCTUREDTEXT

Restructured Text is the default plain text markup language used bySphinx.Docutils provides the basic restructured Text syntax, while Sphinx extends this to support additional functionality. Since rest was designed tobe a simple, unobtrusive markup language, this will not take too long. The primary goal of restructured Text is to define and implement a markup syntax to be used in Python docstrings and alternative documentation domains, that is legible and easy, however powerful enough for non-trivial use. The meant purpose of the markup is that the conversion of restructured Text documents into helpful structured knowledge formats

SPHINX

Sphinx could be a tool that creates it simple to form intelligent and exquisite documentation. it had been originally created for the Python documentation, and its wonderful facilities for the documentation of software system comes. the subsequent options ought to be highlighted:

1. Output formats: HTML, LaTeX (for printable PDF versions), ePub, Texinfo, manual pages, plain text.
2. In depth cross-references: linguistics markup and automatic links for functions, classes, citations, wordbook terms and similar items of knowledge.
3. Hierarchical structure: easy definition of a document tree, with automatic links to siblings, parents, and children
4. Automatic indices: general index as well as a language-specific mod- ule indices.
5. Code handling: automatic highlighting using the Pygments highlighter.

Sphinx needs to be told a language referred to as Rest or RST that stands for restructured Text. The advantage of the remainder syntax resides in its human- legible nature. Taking notes in Rest language additionally|is additionally} convenient since it remains legible however will be also taken later into nice layout (pdf, html). Rest and Sphinx provides an upscale syntax fabricated from directives. Sphinx comes with several of them additionally to straightforward ones provided by the remainder language. For instance, we can include LaTeX code for our equations. Yet, there are many directives and extensions. Sphinx is a Python system to convert .rst files to .html files. After running the sphinx, a HTML file would be made in the same directory as ofrest file. The code of HTML is complicated than normal HTML code written by a person since it is auto generated.

TRANSLATOR FROM RST TO REVEAL.JS

The extension generates Reveal.js presentation from standard reStructuredText.It include these features. Custom builder translates from Rest to reveal.js style HTML.

WEB APPLICATION

Runestone books may be in-built 2 modes: Mode one (dynamic) is meant so books square measure served with the support of the Book server and its associated arthropod genus. Mode 2 may be a standalone mode that does not save any knowledge, need any logins, and permits the user serve content as an easy static web site. a whole Runestone system consists of many servers as shown in

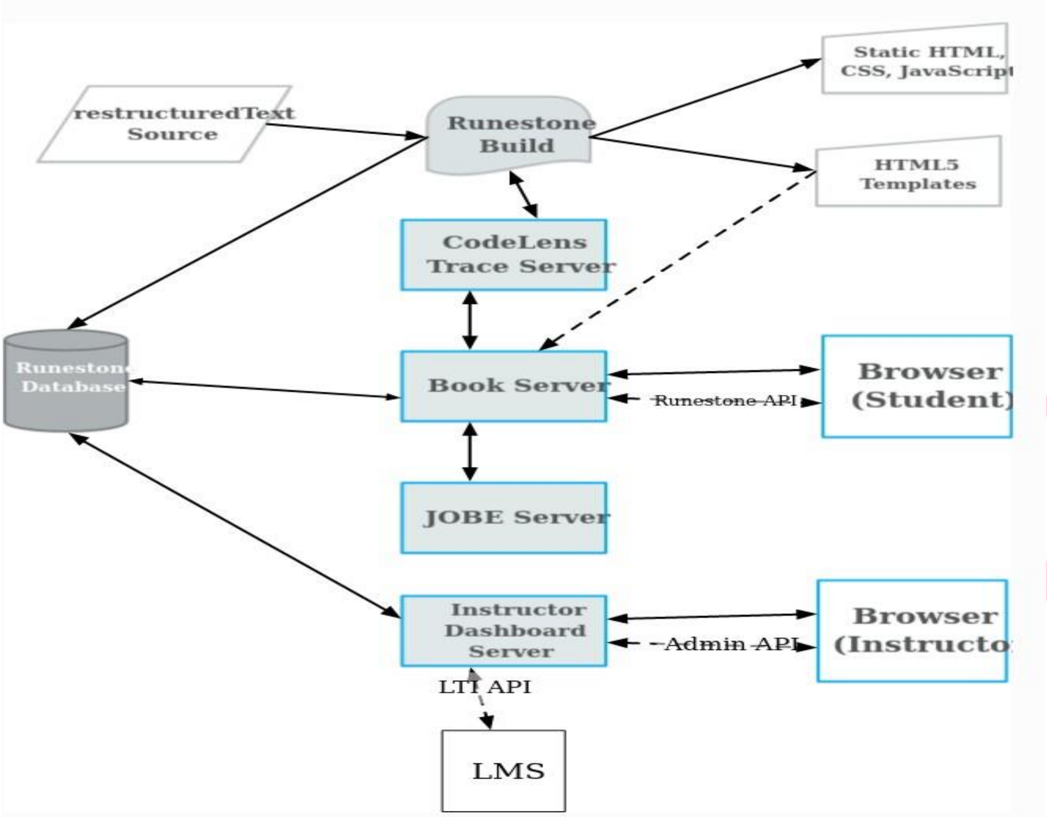


Figure 4.5. In this section we will briefly describe the major components, thegray items in the diagram.

The book server routes students to the pages for their course. In generalrouting involves two important concepts:

1. Base Course: All runestone courses are based around a textbook.
2. Bespoke Course: A particular course at an institution will have a name chosen by the instructor, something like python-intro-f20 for example, the python-intro-f20 course is the name used by the students to access their courses.

All courses map to a base course and therefore the hypertext markup language for that base course is served to students beside some configuration variables inserted for every course at the time the page is served. this enables USA to own one static copy of every book however serve the book for several individual courses.

Results and Discussions

We have designed and constructed a tool for interactive presentation. Aninteractive presentation means the reader will interact with it: while reading the presentation we can ask questions like multiple choice and short answers, after answering the question the review of the answer can be given in the presentation, so that the reader can check her answers. While watching a presentation, readers will answer the question given in the presentation and the answers given by the reader will be collected. In this way, the instructor comes to know about the how much progress the reader has made in listening the presentation, and how many questions they have answered. In an interactive presentation, apart from the question and answers, there can also be programming questions. It means instead of just watching the presentation, the presentation can have pieces of code that the reader can change and/orrun.

Our project started with Runestone interactive books and incorporatedits interactive features into a presentation tool, namely reveal js, and trans- form it into an interactive presentation tool. We have extended Sphinx to include tags for slides which when translated become HTML slides in Reveal.js. Sphinx is a tool to convert Rest document into HTML document. Converting presentation marked in Rest to Reveal.js presentation marked in HTML was carried out by the Sphinx extension. This enables us to incorporate inter- active widgets in the Reveal.js HTML presentation.

Conclusions

With the tool developed, a presentation can be served as web pages synchronously to a group of users, controlled by the presenter. In the web-based presentation, we can execute programs, written in Python, in the browser itself, and let the users change the programs and run them. It collects the responses of the users for the questions and gives feedback, without contacting the server. It sends the collected responses to the server for analysis of the effectiveness of the presentation.

References

[1] Brad Miller and David Ranum. *Runestone Interactive: Tools for Creating Interactive Course Materials*. In Proceedings of the first ACM conference on Learning @ scale conference (L@S 2014). Association for Computing Machinery, New York, NY, USA, 213–214. 2014.

[2] Ericson B, Guzdial M and Morrison B. *Analysis of Interactive Features Designed to Enhance Learning in an eBook*. Proceedings of the EleventhAnnual International Conference on International Computing Education Research, (169-178). 2015.

[3] Kerttu Pollari-Malmi, Julio Guerra, Peter Brusilovsky, Lauri Malmi, and Teemu Sirkia. *On the Value of Using an Interactive Electronic Text-book in an Introductory Programming Course*. In Proceedings of the 17th Koli Calling International Conference on Computing Education Re- search (Koli Calling 2017). Association for Computing Machinery, New York, NY, USA, 168–172. 2017.

[4] Bradley N Miller and David L Ranum. *Beyond PDF and ePub: Toward an Interactive Textbook*. In Proceedings of the 17th ACM annual con- ference on Innovation and technology in computer science education(ITiCSE 2012). Association for Computing Machinery, New York, NY, USA, 150–155. 2012.

[5] Lechtenborger. *Emacs-reveal: A Software Bundle to Create OER Presentations*. Journal of Open-Source Education, 2(18), 50, https://doi.org/10.21105/jose.00050. 2019.

[6] Barbara Ericson and Bradley Miller. *Runestone: A Platform for Free, On-line, and Interactive Ebooks*. In Proceedings of the 51st ACM Technical Symposium on Computer Science Education (SIGCSE 2020). As- sociation for Computing Machinery, New York, NY, USA, 1240. 2020.

[7] Bradley Miller, Paul Resnick, and Barbara Ericson. *Using and Customizing Open-Source Runestone Ebooks for Computer*

Science Classes. In Proceedings of the 2017 ACM SIGCSE Technical Symposium on Computer Science Education (SIGCSE '17). Association for Computing Machinery, New York, NY, USA, 741. 2017.

[8] Lauren E Margulieux, Barbara J Ericson, and Jochen Rick. *Solving Par- sons Problems Versus Fixing and Writing Code*. In Proceedings of the 17th Koli Calling International Conference on Computing Education Re- search (Koli Calling 2017). Association for Computing Machinery, New York, NY, USA, 20–29. 2017.

[9] Kantwon Rogers, Barbara J Ericson, Miranda Parker, Briana Morrison, and Mark Guzdial. *Identifying Design Principles for CS Teacher Ebooks through Design-Based Research*. In Proceedings of the 2016 ACM Conference on International Computing Education Research (ICER '16).Association for Computing Machinery, New York, NY, USA, 191–200. 2016.

[10] Steven Moore,Barbara Ericson, Briana Morrison, and Mark Guzdial. *Us- ability and Usage of Interactive Features in an Online Ebook for CS Teachers*. In Proceedings of the Workshop in Primary and Secondary Computing Education (WiPSCE '15). Association for Computing Machinery, New York, NY, USA, 111–120. 2015.