

---

**FUNCTIONAL  
PROGRAMMING  
CS-IS-2010-1  
MIDTERM EVALUATION  
PROJECT REPORT**

**Haskell Web Scraper**

**Saptarishi Dhanuka**

**Ashoka University**

# Contents

<b>1</b>	<b>Problem Statement and Requirements</b>	<b>3</b>
1.1	Problem Description . . . . .	3
1.2	Requirements . . . . .	3
<b>2</b>	<b>Specifications</b>	<b>4</b>
<b>3</b>	<b>Design and Architecture</b>	<b>5</b>
<b>4</b>	<b>Choice of Tools, Platforms, and Languages</b>	<b>6</b>
<b>5</b>	<b>Test Plan</b>	<b>7</b>
<b>6</b>	<b>Prototype Implementation Details</b>	<b>8</b>
<b>7</b>	<b>Plan for Completion</b>	<b>9</b>

# 1 Problem Statement and Requirements

## Assigned Project Statement

Develop a scraper using Haskell to extract text and code snippets separately.

1. **Input:** Scrape the text and code snippets from the given text **source**
2. **Output:** A Word document containing the text and **.txt** file containing the code.
3. **Method:** Write the algorithm to scrape (you can use the **tagsoup** library) and all the input-output facilities using Haskell. Do not use any other language.

## 1.1 Problem Description

The given web page is made of text and code snippets, which we need to scrape and extract separately into a **.docx** file containing the text portions and a **.txt** file which has the code snippets.

For this, we need to fetch the given web page, parse and analyze its HTML structure to identify the HTML tags of the code snippets and the tags of the rest of the text, so that we can effectively separate them into different documents.

## 1.2 Requirements

1. The scraper will be written entirely in Haskell
2. The scraper will get all text of the given page and write it into a **.docx** file.
3. The scraper will get all the code snippets of the given page and write it into a **.txt** file.

## 2 Specifications

1. The scraper will use Haskell HTTP libraries for fetching the contents of the given web page.
2. The scraper will separate the code snippets from the rest of the textual content using the **tagsoup** library
3. The scraper will write the text into a Word document and the code snippets into a `.txt` file mainly using the **tagsoup** and **pandoc** libraries, along with some standard Haskell libraries.

## **3 Design and Architecture**

## **4 Choice of Tools, Platforms, and Languages**

## 5 Test Plan

## **6 Prototype Implementation Details**



## **7 Plan for Completion**