# **Task1 Documentation**

VXD240002, Dacharla Venkata Abhiram

## Introduction

I successfully completed Task1 by extracting course information from an HTML page (coursebook2025.html) and generating a SQL script to create and populate a database table. This documentation outlines the steps I took to achieve this.  
  
**Project Setup**

I used Xampp to complete this task. I created a /cs6314/task1 folder in the htdocs directory. Within this folder, I created:

* task1.html: The main HTML file containing the course content
* js folder: Contains task1.js, the JavaScript file for data extraction
* exports folder: Stores all the exported data files in various format

A screenshot of a computer

AI-generated content may be incorrect.

## Setting Up the HTML File

I created a new HTML file called task1.html and included the content from coursebook.html within it by creating a div named coursebook-content. This allowed me to link my JavaScript file (task1.js) to the HTML page and perform data extraction directly from the browser.

A screenshot of a computer program

AI-generated content may be incorrect.

## Data Extraction Process

I wrote JavaScript functions to extract each column of information from the HTML table which is in section-list div in the original HTML file:

A screenshot of a computer

AI-generated content may be incorrect.

## Extracting Year-Semester

I created a function extractYearSemester to get the year and semester information from the first column of each table row. This function checks if the cell exists and returns the trimmed text or "N/A" if the information isn't available.

## Extracting Course Number and Section

I implemented a function extractCourseNumberAndSection to extract the course number and section from the second column. This function looks for a link element within the cell and returns the text of the link or "N/A" if not found.

## Extracting Course Title

I developed a function extractCourseTitle to get the course title from the fourth column. Similar to previous functions, it returns the trimmed text of the cell or "N/A" if empty.

## Extracting Instructor Details

I created a complex function extractInstructorDetails to extract instructor names and their corresponding NetIDs from the fifth column.

This function:

* Finds all link elements in the cell.
* Extracts the instructor name from the link text.
* Extracts the NetID from the link's href attribute.
* Handles cases where multiple instructors are listed.
* Returns "N/A" for name and NetID if no information is available.

## Extracting Schedule and Location

I implemented a function extractScheduleAndLocation to extract schedule and location information from the sixth column.

This function:

* Splits the cell text into lines
* Separates the schedule (all lines except the last) from the location (last line)
* Handles cases where only schedule or location information is present
* Returns "N/A" for missing information

## Adding Timestamp Fields

I included two additional fields for each course:

* time\_created: Automatically set to the current timestamp when a record is created
* time\_updated: Automatically updated to the current timestamp when a record is modified

## Generating the SQL File

When task1.html is opened in a browser, my JavaScript code:

* Extracts all course information using the functions I created
* Generates a SQL script that creates a table and inserts all extracted data
* Downloads the SQL file automatically

The SQL script creates a table named "course1" with appropriate data types and constraints, including primary keys and default timestamp values.

When task1.html is run in browser we can download the sql file for the extrated table.

A computer screen shot of a computer screen

AI-generated content may be incorrect.

## Database Implementation

I created a database named "cs6314" using phpMyAdmin and imported the generated task1.sql file. This created a table named "course1" with 236 entries containing all the extracted course information.

A screenshot of a computer

AI-generated content may be incorrect.

Here is the screenshot showing all the data extracted into coure1 table

A screenshot of a computer

AI-generated content may be incorrect.

## Data Export

I exported the "course1" table in various formats and saved them in the "/task1/exports" directory:

* XML format: task1out.xml
* JSON format: task1out.json
* PHP array format: task1out.php
* SQL format: task1out.sql

A screenshot of a computer

AI-generated content may be incorrect.

## Similarly I downloaded in other formats also.

## A screenshot of a computer AI-generated content may be incorrect.

## Conclusion

I successfully implemented Task1 by extracting course information from an HTML page and generating a SQL script. The solution demonstrated the ability to handle various data formats and edge cases, resulting in a properly structured database table with 236 entries. The exported data files in multiple formats further validated the completeness of the extraction process.