



# INSTITUTE OF AERONAUTICAL ENGINEERING (AUTONOMOUS)

Dundigal - 500 043, Hyderabad, Telangana

## Complex Problem-Solving Self-Assessment Form

1	Name of the Student	K AARON
2	Roll Number	25951A6601
3	Branch and Section	CSE-(AI&ML) - A
4	Program	B. Tech
5	Course Name	Front-End Web Development
6	Course Code	ACSE04
7	Please tick (✓) relevant Engineering Competency (ECs) Profiles	
EC	Profiles	(✓)
EC 1	Ensures that all aspects of an engineering activity are soundly based on fundamental principles - by diagnosing, and taking appropriate action with data, calculations, results, proposals, processes, practices, and documented information that may be ill-founded, illogical, erroneous, unreliable or unrealistic requirements applicable to the engineering discipline	✓
EC 2	Have no obvious solution and require abstract thinking, originality in analysis to formulate suitable models.	✓
EC 3	Support sustainable development solutions by ensuring functional requirements, minimize environmental impact and optimize resource utilization throughout the life cycle, while balancing performance and cost effectiveness.	
EC 4	Competently addresses complex engineering problems which involve uncertainty, ambiguity, imprecise information and wide-ranging or conflicting technical, engineering and other issues.	✓
EC 5	Conceptualises alternative engineering approaches and evaluates potential outcomes against appropriate criteria to justify an optimal solution choice.	✓

	EC 6	Identifies, quantifies, mitigates and manages technical, health, environmental, safety, economic and other contextual risks associated to seek achievable sustainable outcomes with engineering application in the designated engineering discipline.	
	EC 7	Involve the coordination of diverse resources (and for this purpose, resources include people, money, equipment, materials, information and technologies) in the timely delivery of outcomes	
	EC 8	Design and develop solution to complex engineering problem considering a very perspective and taking account of stakeholder views with widely varying needs.	✓
	EC 9	Meet all level, legal, regulatory, relevant standards and codes of practice, protect public health and safety in the course of all engineering activities.	
	EC 10	High level problems including many component parts or sub-problems, partitions problems, processes or systems into manageable elements for the purposes of analysis, modelling or design and then re-combines to form a whole, with the integrity and performance of the overall system as the top consideration.	✓
	EC 11	Undertake CPD activities to maintain and extend competences and enhance the ability to adapt to emerging technologies and the ever-changing nature of work.	✓
	EC 12	Recognize complexity and assess alternatives in light of competing requirements and incomplete knowledge. Require judgement in decision making in the course of all complex engineering activities.	✓
8	Please tick (✓) relevant Course Outcomes (COs) Covered		
	CO	Course Outcomes	(✓)
	CO 1	Describe language basics like alphabet, strings, grammars, productions, derivations, and Chomsky hierarchy, construct DFA, NFA, and conversion of NFA to DFA, Moore and Mealy machines and interpret differences between them.	✓
	CO 2	Recognize regular expressions, formulate, and build equivalent finite automata for various languages.	✓
	CO 3	Identify closure, and decision properties of the languages and prove the membership.	✓
	CO4	Demonstrate context-free grammars, check the ambiguity of the grammar, and design equivalent PDA to accept the context-free languages.	
	CO 5	Uses mathematical tools and abstract machine models to solve complex problems.	✓
	CO 6	Analyze and distinguish between decidable and undecidable problems.	✓

		Number of Videos	Viewing time in Hours
9	Course ELRV Video Lectures Viewed	-	-
10	Justify your understanding of WK1	-	-
11	Justify your understanding of WK2 – WK9	-	-
12	How many Wks from WK2 to WK9 were implanted?  Mention them	-	-



Date: 05-12-2025

Signature of the Student

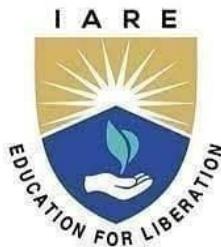


**Invoicer**  
**A Project Report submitted in**  
**partial fulfilment of the requirements**  
**for the award of the degree of**  
**Bachelor of Technology in**  
**CSE (Artificial Intelligence & Machine Learning)**

**By**

**K Aaron**

**25951A6601**



**Department of CSE (Artificial Intelligence & Machine Learning)**

**INSTITUTE OF AERONAUTICAL ENGINEERING**  
**(Autonomous)**

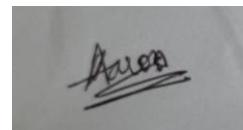
**Dundigal, Hyderabad – 500 043, Telangana**

**November, 2025**

## **DECLARATION**

I certify that

- a. The work contained in this report is original and has been done by me under the guidance of my supervisor (s).
- b. The work has not been submitted to any other Institute for any degree or diploma.
- c. I have followed the guidelines provided by the Institute for preparing the report.
- d. I have conformed to the norms and guidelines given in the Code of Conduct of the Institute.
- e. Whenever I have used materials (data, theoretical analysis, figures, and text) from other sources, I have given due credit to them by citing them in the text of the report and giving their details in the references. Further, I have taken permission from the copyright owners of the sources, whenever necessary.



**Place: Hyderabad**

**Signature of the Student**

**Date: 05-12-2025**

## **CERTIFICATE**

This is to certify that the project report entitled **Invoicer** submitted by **K Aaron** to the Institute of Aeronautical Engineering, Hyderabad in partial fulfilment of the requirements for the award of the Degree Bachelor of Technology in **CSE - (ARTIFICIAL INTELLIGENCE & MACHINE LEARNING)** is a Bonafide record of work carried out by his guidance and supervision. The Contents of this report, in full or in parts, have not been submitted to any other Institute for the award of any Degree.

**Supervisor**

**Date: 05-12-2025**

**Head of the Department**

**Principal**

## **APPROVAL SHEET**

This project report entitled **Invoicer** submitted by **K Aaron** is approved for the award of the Degree Bachelor of Technology in Branch **CSE (Artificial Intelligence & Machine Learning)**.

**Examiner**

**Supervisor(s)**

**Principal**

**Date: 05-12-2025**

**Place: Hyderabad**

## **ACKNOWLEDGEMENT**

The satisfaction that accompanies the successful completion of any task would be incomplete without introducing the people who made it possible and whose constant guidance and encouragement crowns all efforts with success.

I am extremely grateful and express my profound gratitude and indebtedness to my project guide **Mr. Vidyasagar Vidapu, Assistant Professor, Department of CSE – (Artificial Intelligence and Machine Learning)**, for his kind help and for giving me the necessary guidance and valuable suggestions for this project work.

I am grateful to **Dr. M. Purushotham Reddy, Professor and Head of the Department**, Department of **CSE (Artificial Intelligence & Machine Learning)**, for extending his support to carry on this project work. I take this opportunity to express my deepest gratitude to one and all who directly or indirectly helped me in bringing this effort to present form.

I express my sincere gratitude to **Dr. L. V. Narasimha Prasad, Professor and Principal** who has been a great source of information for my work.

I thank our college management and respected **Sri M. Rajashekhar Reddy, Chairman, IARE, Dundigal** for providing me with the necessary infrastructure to conduct the project work.

I take this opportunity to express my deepest gratitude to one and all who directly or indirectly helped me in bringing this effort to present form.

## ABSTRACT

Invoicing is an essential component of business operations, yet many individuals and small enterprises struggle with inefficient, inconsistent, or time-consuming invoice creation methods. Traditional approaches such as manual formatting or spreadsheet-based invoicing often lead to errors, lack of professionalism, and difficulties in record management. **Invoicer** is a lightweight, front-end-based invoice generator designed to address these challenges by providing a simple, efficient, and user-friendly platform for creating professional client invoices.

The system enables users to input client details, add service or product items, automatically calculate totals, taxes, and discounts, and select from multiple pre-designed templates. With real-time updates, clean layouts, and localStorage-supported data persistence, users can quickly generate and download polished invoices without requiring backend support or advanced technical knowledge. The intuitive design and automated formatting significantly enhance the speed, accuracy, and consistency of invoice creation.

Invoicer demonstrates how HTML, CSS, and JavaScript can be effectively combined to deliver a functional business tool that streamlines invoicing processes, supports professional documentation, and improves overall workflow efficiency. The project serves as a practical solution for freelancers, small business owners, and service providers seeking a simple yet reliable invoice-generation platform.

## **CONTENTS**

<b>Name of Contents</b>	<b>Page No.</b>
Title Page	I
Declaration	II
Certificate	III
Approval Sheet	IV
Acknowledgement	V
Abstract	VI
Contents	VII
Chapter 1- Introduction	8-9
1.1 Problem Statement	8
1.2 Introduction	8
1.3 Requirements	9
1.4 Prerequisites	10
1.5 Technologies used	10

Chapter 2 - Review of Relevant Literature	11-12
Chapter 3- Methodology	13-14
Chapter 4- Results and Discussions	15-16
Chapter 5- Conclusions and Future Scope	17-18
5.1 Conclusion	17
5.2 Future Scope	18
References	19

# CHAPTER 1 INTRODUCTION

## 1.1 Problem Statement

Small businesses, freelancers, and service providers often face difficulties managing invoices due to inconsistent templates, manual formatting, and repetitive data entry. Traditional invoicing methods using spreadsheets or handwritten bills lack professionalism, accuracy, and automation. These approaches also make it harder to track payments, maintain records, and generate new invoices quickly.

Although various invoicing tools exist, many are either expensive, overly complex, or require backend integrations. There remains a need for a **simple, lightweight, front-end based invoice generator** where users can instantly create, customize, download, and store invoices without technical knowledge.

**Invoicer** aims to fill this gap by providing a clean, user-friendly interface where users can quickly enter client details, item lists, and pricing information. With built-in templates and auto-calculation features, users can generate polished professional invoices with minimal effort. This tool helps streamline business operations, save time, and maintain consistency across client documents.

## 1.2 Introduction

**Invoicer** is a web-based front-end application designed to help individuals and businesses generate professional invoices with ease. It eliminates the need for complex accounting tools by offering a straightforward interface that focuses on essential invoice creation features.

Users can input client details, add items or services, specify quantities and rates, and choose from pre-designed invoice templates. Totals, taxes, and discounts are calculated automatically in real time, ensuring accuracy and reducing manual errors.

The application allows users to:

- Create invoices quickly
- Customize appearance using invoice templates
- Download invoices as PDF
- Save drafts using localStorage
- Maintain consistent branding

With a responsive and intuitive UI, **Invoicer** makes the invoicing process simple, fast, and efficient for businesses of any size.

## **1.3 Requirements**

### **1. Functional Requirements (FrontEnd)**

#### **1.1 Invoice Creation**

- FR1: Users must be able to input client information (name, address, email).
- FR2: Users must be able to add multiple invoice items (description, quantity, rate).
- FR3: The system must auto-calculate totals, taxes, and discounts.
- FR4: Users must be able to delete or update invoice items.
- FR5: The invoice must display a professional layout with proper alignment and formatting.

#### **1.2 Templates**

- FR6: Users must be able to switch between multiple built-in invoice templates.
- FR7: Template selection should update the preview instantly.
- FR8: Templates must support business branding (logo, business name, contact details).

#### **1.3 Export & Storage**

- FR9: Users must be able to download the invoice as a PDF (front-end based).
- FR10: Invoice data must be saved automatically in localStorage.
- FR11: Users must be able to load saved drafts and continue editing.

## **2. Non-Functional Requirements**

- NFR1: UI must be responsive across desktop, tablet, and mobile devices.
- NFR2: Interface must be simple, intuitive, and business-friendly.
- NFR3: Auto-calculations should update instantly without delay.
- NFR4: Templates and design must follow consistent typography and spacing.
- NFR5: Data should persist reliably using localStorage.
- NFR6: Application must maintain accurate values even after multiple edits.

## **1.4 Pre-requisites**

### **1. Technical**

- Knowledge of HTML5 for structure and form design
- CSS3 for layout, template styling, and invoice formatting
- JavaScript for calculations, dynamic updates, and item manipulation
- DOM manipulation for rendering templates and updating previews
- Knowledge of LocalStorage for draft saving
- Optional: Familiarity with jsPDF or HTML-to-PDF libraries to generate PDFs

### **2. Tools**

- Code Editor (VS Code recommended)
- Modern web browser for testing
- Optional design tool (Figma/Canva) for template planning
- GitHub for version control and deployment

### **3. User**

- Basic understanding of invoice fields
- Ability to input numeric values (rates, quantities)
- Familiarity with downloading documents

## **1.5 Technologies Used**

### **1. HTML5**

- Form fields for invoice inputs
- Layout for templates and preview area
- Semantic tags for readability and structure

### **2. CSS3**

- Styling invoice templates
- Professional business layouts
- Responsive design for multiple devices
- Typography and spacing for clean invoice presentation

### **3. JavaScript**

- Dynamic calculations (subtotals, taxes, discounts, totals)
- DOM manipulation for template switching
- Adding/removing invoice items
- Saving and loading data from localStorage

### **4. jsPDF / html2canvas (Optional)**

- Export invoices as downloadable PDF

## 5. Version Control

- Git & GitHub for hosting and updates

# CHAPTER 2

## REVIEW OF RELEVANT LITERATURE

Studies in small business management highlight that efficient billing processes improve cash flow and client satisfaction. Research also shows that clear, well-formatted invoices reduce payment delays and misunderstandings between clients and service providers.

Front-end invoice generators are increasingly preferred due to their simplicity and accessibility. Many literature sources emphasize the importance of:

- Auto-calculated totals
- Professional templates
- Fast invoice creation
- Customizability

Modern UI/UX design principles influence how invoice tools present information. Clean layouts, minimal design, and structured sections reduce cognitive load and make invoices easier to understand.

Client-side storage mechanisms like `localStorage` have been praised in productivity tools for enabling offline access and quick retrieval of drafts without backend dependencies. These findings align with **Invoicer's** lightweight front-end architecture.

## **CHAPTER 3 METHODOLOGY**

The development process followed these steps:

### **1. UI Planning & Template Design**

- Created mockups for invoice templates
- Structured input fields and preview layout

### **2. Front-End Development**

- Implemented HTML structure for invoice forms and templates
- Used CSS to style invoices professionally
- Implemented JavaScript functions for:
  - Adding invoice items
  - Performing automatic calculations
  - Switching templates
  - Saving and loading drafts

### **3. Testing**

- Verified accuracy of calculations
- Tested responsiveness across screen sizes
- Ensured templates rendered properly in PDF format

### **4. Deployment**

- Uploaded to GitHub Pages
- Maintained version history using Git

This methodology ensures a functional, efficient, and visually polished invoice generator suitable for business use.

```
<!DOCTYPE html>

<html>
<head>
<title>Ultra Simple Invoicer</title>
<style>
body { font-family: Arial; padding:20px; }

input { width:100%; padding:8px; margin:5px 0; }

table { width:100%; border-collapse:collapse; margin-top:10px; }

th, td { border:1px solid #ccc; padding:8px; }

button { padding:8px 12px; background:black; color:white; border:none;
cursor:pointer; }

</style>
</head>
<body>

<h2>Ultra Simple Invoicer</h2>

<input id="biz" placeholder="Your Business Name">
<input id="client" placeholder="Client Name">

<h3>Add Item</h3>
```

```

<input id="d" placeholder="Description">
<input id="q" type="number" placeholder="Qty">
<input id="r" type="number" placeholder="Rate">
<button onclick="add()">Add Item</button>

<table>

<thead><tr><th>Description</th><th>Qty</th><th>Rate</th><th>Total</th>
</tr></thead>

<tbody id="rows"></tbody>

</table>

<h3>Total: ₹<span id="t">0</span></h3>

<script>

let items=[];

function add(){

if(!d.value || !q.value || !r.value) return;

items.push({d:d.value,q:+q.value,r:+r.value});

d.value=q.value=r.value="";
}

render();

```

```
}

function render(){

rows.innerHTML=""; let sum=0;

items.forEach(i=>{

let tot=i.q*i.r; sum+=tot;

rows.innerHTML+=`<tr><td>${i.d}</td><td>${i.q}</td><td>${i.r}</td><td>${t
ot}</td></tr>`;

});

t.innerText=sum;

}

</script>

</body>

</html>
```

## OUTPUT:

### Ultra Simple Invoicer

Your Business Name

Client Name

Add Item

Description

Qty

Rate

Add Item

Description	Qty	Rate	Total

Total: ₹0

## CHAPTER 4

### RESULTS AND DISCUSSIONS

Testing confirmed that **Invoicer** meets its objectives of enabling quick and accurate invoice generation. The auto-calculation feature significantly reduces manual errors, especially in subtotal, tax, and discount entries.

Users found template switching smooth and intuitive. The clean invoice layouts improved readability and professionalism, making the tool suitable for real client interactions. The PDF export feature allowed invoices to be shared easily.

LocalStorage proved effective in storing drafts, allowing users to return later and resume editing without losing progress.

Overall, the application performed consistently across devices and browsers, validating the reliability of the front-end implementation.

# **CHAPTER 5**

## **CONCLUSION AND FUTURE SCOPE**

### **5.1 Conclusions**

**Invoicer** successfully demonstrates how front-end web technologies can be used to build a practical business tool. It provides an efficient way to generate client invoices using customizable templates and automated calculations, all within a user-friendly interface. The use of JavaScript for real-time updates and localStorage for data persistence made the tool reliable and convenient.

The project highlights the importance of simplicity, accuracy, and professional design in business documentation tools.

### **5.2 Future Scope**

Potential improvements include:

- Cloud database integration for multi-device syncing
- Exporting invoices in additional formats (DOCX, XLSX)
- Adding signature support (digital sign)
- Payment status tracking dashboard
- Multi-currency support for international clients
- AI-based invoice suggestions and templates
- Option to email invoices directly from the interface

With these upgrades, Invoicer can evolve into a comprehensive invoicing system suitable for small and medium-scale enterprises.