

INTERNSHIP REPORT ON SAVE++, AN INCOME TAX CALCULATOR

(March 2022-June 2022)

Internship report submitted in partial fulfilment of the requirement for the degree
of Bachelor of Technology

in

Computer Science and Engineering

By

Apurva Dubey (181209)

Under the supervision of

Dr. Ruchi Verma

to



Department of Computer Science & Engineering and Information Technology
**Jaypee University of Information Technology Waknaghat, Solan-173234, Himachal
Pradesh**

DECLARATION

I hereby declare that the work presented in this report entitled “**Save++**” in partial fulfilment of the requirements for the award of the degree of **Bachelor of Technology in Computer Science and Engineering/Information Technology** submitted in the department of Computer Science & Engineering and Information Technology, Jaypee University of Information Technology Waknaghat is an authentic record of my own work carried out at **Hashedin By Deloitte**-A service based company over a period from **March 2022 to May 2022** under the supervision of **Dr. Ruchi Verma** (Assistant Professor (SG), Dept. CSE & IT) and **Vatsal Mahendra** (Functional Analyst at Hashedin By Deloitte , Dept. Product Delivery).

The matter embodied in the report has not been submitted for the award of any other degree or diploma.

Apurva Dubey, 181209

This is to certify that the above statement made by the candidate is true to the best of my knowledge.

Dr. Ruchi Verma
Assistant Professor (SG)
Dept. CSE & IT



Vatsal Mahendra
Functional Analyst
Dept. Product Delivery

ACKNOWLEDGEMENT

Firstly, I express my heartiest thanks and gratefulness to Almighty God for his divine blessing that makes it possible to complete the project work successfully.

I am really grateful and wish my profound indebtedness to Supervisor **Dr. Ruchi Verma, Assistant Professor(SG)**, Department of CSE & IT, Jaypee University of Information Technology, Wakhnaghat. The keen interest of my supervisor, her endless patience, scholarly guidance, continual encouragement, constant and energetic supervision, constructive criticism, valuable advice, reading many inferior drafts, and correcting them at all stages have made it possible to complete this project.

I would like to express my heartiest gratitude to **Dr. Ruchi Verma**, Department of CSE, for her kind help to finish my project.

I would also generously welcome each one of those individuals who have helped me straightforwardly or in a roundabout way in making this project a win. In this unique situation, I might want to thank the various staff individuals, both educating and non-instructing, which have developed their convenient help and facilitated my undertaking.

Finally, I must acknowledge with due respect the constant support and patience of my parents.

Apurva Dubey (181209)

TABLE OF CONTENTS

Sr. No.	Contents	Page No.
	Certificate	2
	Acknowledgment	3
	List of Abbreviations	5
	List of Figures	5
	List of Annotations	5
	Abstract	6
1.	Introduction I. Introduction to the product II. Purpose III. Scope IV. Product Perspective	7
2.	Product Characteristics and its use cases I. How to use the product II. How to calculate tax III. Use cases IV. Constraints V. Architecture Diagram	11
3.	Tools And Technologies Used I. Github II. Postman III. MySql IV. Typescript V. React VI. Spring Boot VII. CI/CD VIII. Figma	13
4.	Storyline of Product	23
5.	Performance Analysis I. Testing in React js II. Testing in Spring Boot III. Results	27
6.	Conclusion	34
7.	References	35

LIST OF ABBREVIATIONS

1. STS: Spring Tool Suite
2. JPA: Java Persistence API
3. SQL: Structured Query Language
4. DB: database
5. HTTP: hypertext transfer protocol
6. API: Application Programming Interface
7. VCS: version control system
8. OS: operating system
9. CRUD: create, read, update, delete

LIST OF ANNOTATIONS

1. `@SpringBootApplication`
2. `@Entity`
3. `@Table`
4. `@Id`
5. `@GeneratedValue`
6. `@Controller`
7. `@RequestMapping`
8. `@Autowired`
9. `@GetMapping`
10. `@Repository`
11. `@Query`
12. `@Service`
13. `@SpringBootTest`
14. `@Test`

LIST OF FIGURES

1. Fig.1 Architecture Diagram
2. Fig.2 Roadmap of the product
3. Fig.3 Backend Repository
4. Fig.4 Backend Deployment Workflows
5. Fig.5 Frontend Deployment Records
6. Fig.6 Frontend Repository
7. Fig.7 Commits History
8. Fig.8 Income tax Calculator form
9. Fig.9 Tax Report
10. Fig.10 Tablet View Of Application
11. Fig.11 Mobile View

ABSTRACT

The term income tax refers to a type of tax that governments impose on income generated by businesses and individuals within their jurisdiction. By law, taxpayers must file an income tax return annually to determine their tax obligations. Income taxes are a source of revenue for governments.

Income tax is a direct tax that a government levies on the income of its citizens. The Income Tax Act, 1961, mandates that the central government collect this tax. The government can change the income slabs and tax rates every year in its Union Budget.

Individual income tax is also referred to as personal income tax. This type of income tax is levied on an individual's wages, salaries, and other types of income. This tax is usually a tax the state imposes.

From Financial year 2020-21, taxpayers (Individuals and HUFs) have an option to choose between the Old Tax regime and the new Tax Regime. The new tax regime is available with lower tax rates and almost zero deductions/exemptions.

Most of the exemptions and deductions are not available under the new tax regime.

There is no single answer to opt for any one as it depends on your annual earnings and component of exemptions and deductions included in it. However, the new tax regime may seem beneficial due to reduced tax rates at various slabs.

The choice is available every year and any regime which is beneficial can be adopted by the individual & HUF (except for those who have income from business or profession).

Individuals who have income from business or profession cannot switch between the new and old regimes every year. If they opt for the new regime, such taxpayers get only one chance in their lifetime to go back to the old regime. Further, once switched back to the existing tax regime, they will not be able opt for the new regime unless their business income ceases to exist.

Income tax is levied on the income earned after considering few deductions. With too many tax exemptions and deductions, calculating tax can be complicated. An income tax calculator is a tool that will help calculate taxes one is liable to pay under the old and new tax regimes.

CHAPTER 1: INTRODUCTION

1.1 Introduction to product

The Save++ is an easy-to-use online tool that helps users to estimate taxes based on their income after the Union Budget is presented and gives the best suggestions to reduce your tax liability. It helps to reduce the long detail filling hassle by autofill feature and observe your taxation history.

1.2 Purpose

The purpose of the project is to help the user in tax calculation and plan their investment accordingly to maximise tax saving. HU Taxers support income tax calculation for individual users working as service employees in the private sector. Moreover, it includes tax saving suggestions and benefits applicable.

1.3 Scope

- Calculate Income Tax for user on the basis of various salary components
- Help user in choosing tax regimes
- Suggest user how to do tax savings

1.4 Product Perspective

The proposed Save++, can be used at any time, and allows creating user profile and autofill salary components for calculating income tax. User profile can be modified in case of change in plans of investment or salary increments.

Software Requirement

- Front end:
 - React JS
- Back end:
 - Spring Boot
- MySql

- Deployment:
 - Google Cloud Platform
 - CI/CD

Hardware Requirement

- 4GB ram
- 1.2 GHz processor
- Intel i5

Infrastructure Requirements

- Front end: Cloud Run with Cloud Registry
- Back end: Cloud Run, Cloud Storage

Functional Requirements

- **Register**
 - **Description** : First the user will have to register/sign up.
 - **Users** : The user has to enter name, email id, password, company name
- **Login**
 - **Input**: use the email id as username and password
 - **Output** : User will be redirected to the User Dashboard of the HU Taxer
- **User Dashboard**
 - **Financial Profile**
 - **Input** : User can use a simple multi-step form to enter his income and investment details along with the suggestions provided in the form for each input component
 - **Output** : Redirected to the next step of form
 - **Design**
 - **Input** : enter details of input fields of a particular step
 - **Output** : Generate a pie chart to represent finance distribution
- **Income Tax Report**
 - **State** : Database contains the updated financial profile of user

- Input : Choose the form from the dashboard.
- Output : Generate the tax to be paid under both the regimes and the scope of maximising tax savings
- **Edit the form**
 - State : Initially, the input fields will be assigned with default values
 - Input : Enter updated values of desired input field
 - Processing : The form will be modified and the changes will be affected to the users
 - Output : Form builder page with the designed form, which can be modified.

Non Functional Requirements

- **Usability Requirement**

The system shall allow the users to access the system from a smartphone, tablet or a desktop with a web browser and an active internet connection. The system uses a web browser as an interface.

Since all users are familiar with the general usage of internet browsing, no special training is required.

The system is user friendly which makes the system easy to use.

- **Availability Requirement**

The system is available 100% for the user and is used 24 hrs a day and 365 days a year. The system shall be operational 24 hours a day and 7 days a week.

- **Efficiency Requirement**

Basic tax calculation details will be autofilled and distribution of the finances across all the components will be updated as the user modifies the data.

- **Accuracy**

The system should accurately calculate the tax liability of the user and help them take informed decisions in deciding how to optimally utilize their capital. The system shall provide 100% access reliability.

- **Performance Requirement**

The information is refreshed depending upon whether some updates have occurred or

not in the application. The system shall respond to the member in not less than two seconds from the time of the request submission. The system shall be allowed to take more time when doing large processing jobs. Responses to view information shall take no longer than 5 seconds to appear on the screen.

- **Reliability Requirement**

The system has to be 100% reliable due to the importance of data and the damages that can be caused by incorrect or incomplete data. The system will run 7 days a week, 24 hours a day.

CHAPTER 2: PRODUCT CHARACTERISTICS AND USE CASES

2.1 How to use HU Taxer for FY 2022-23? (AY 2023-24)?

To know your tax liability with an income tax calculator, simply enter relevant details in the fields:

- Select your age bracket
- Enter your annual income
- Disclose investments and eligible deductions under Sections like:
 - 80C (ELSS funds, PPF, house loan principal repayment, etc.)
 - 80CCD18) (National Pension System)
 - 24B (Home loan interest repayment)
 - 80E (Education loan interest repayment)
 - 806 (donations to charitable institutions)
- Enter HRA, LTA exemptions

Currently the web application services are limited to Individual taxpayers other categories are not covered.

You can enter '0' for fields that are not applicable. Once you go through the steps, you will see your tax payable under the old and new regimes for AY 2023-24 (FY 2022-23).

2.2 How is income tax calculated?

Income tax is calculated on taxable income based on the tax slab applicable. taxable income is derived by adding income from all sources (salary, rent, capital gains, etc.) to get the gross total income and subtracting from this the deductions and exemptions one is eligible for.

2.3 Use Cases

- **The user can enter details in the tax calculator and submit it.**

The user need not enter the same details every time he logs in.
- **The user can decide on using the suggested measures to reduce tax liability.**

The user can modify investment components on the basis of suggestions and can evaluate the tax without any delay

- Detailed comparison of using the old and new tax regime will be shown to make an informed decision.

The user gets help in choosing the regimes on the basis of the tax paid amount displayed on the report step.

2.4 Constraints

Currently the web application services are limited to Individual taxpayers other categories are not covered.

2.5 Architecture Diagram

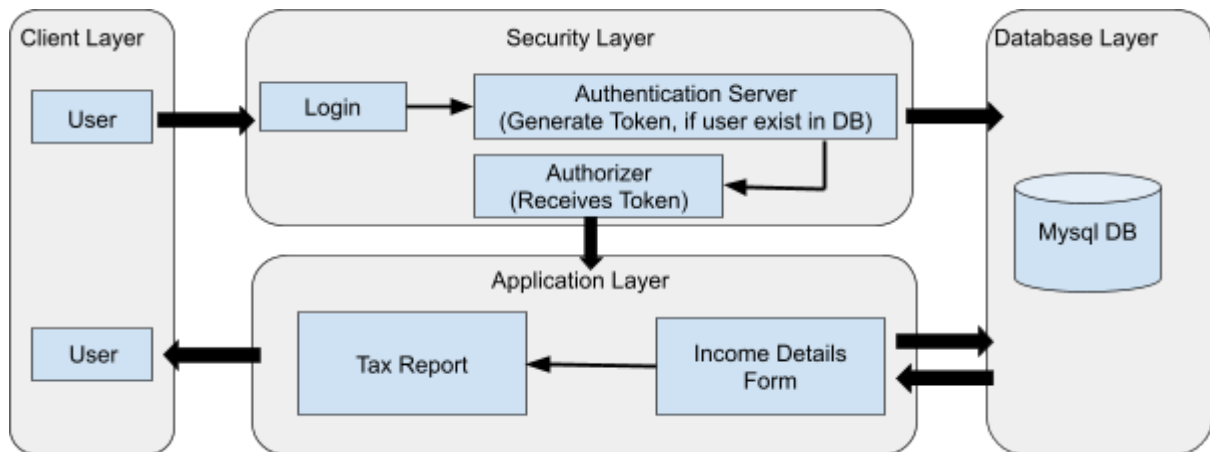


Fig. 1 Architecture Diagram

Chapter 03: TOOLS AND TECHNOLOGIES USED

3.1 Github

Software developers and engineers can use GitHub to build public-facing cloud repositories for free. You can transfer a GitHub repository to your device, add and alter files locally, and then "push" your changes back to the repository, where they will be visible to the public.

3.1.1 Repository

A repository (sometimes shortened as "repo") is a site where all of the files for a project are kept. Each project has its repository, which you can access using a specific URL.

3.1.2 Forking a Repo

When you establish a new project based on an existing project, this is known as "forking." This is a fantastic tool that greatly fosters the development of new applications and projects. If you find a project on GitHub that you'd want to contribute to, you can fork it, make the modifications you want, and then republish it as a new repo. You may quickly add changes to your current fork if the original repository that you forked to create your new project is updated.

3.1.3 Git Commit

You're ready to commit once you've staged the files you want to add. The commit message is important regardless of whether you commit using GitHub Desktop or the command line. Short commit statements that describe your change should be used. The commit messages will lead you through the history of your repository, therefore they should be informative. The following message format can be used in command-line commits:

```
git commit -m "git commit message example"
```

3.1.4 Pull Requests

You've forked a repository, made a fantastic change to the project, and want the original devs to notice it—perhaps even put it in the original project/repository. Create a pull request to accomplish this. The original repository's authors can view your work and decide whether or not to accept it into the official project. When you submit a pull request, GitHub provides an excellent communication channel between you and the primary project's maintainer.

3.2 Postman

Postman is a well-known API testing tool. APIs may be simply created, tested, shared, and documented with the help of this tool. Postman is an Application Program Interface (API) testing tool that may be used to create, test, develop, change, and document APIs. It's a basic graphical user interface for executing and analysing HTTP requests and responses. When utilising Postman for testing, you don't need to write any HTTP connection code. Instead, we use Postman to connect with the API and create test suites called collections. Almost every feature a developer can require is included in this program. This tool can do GET, POST, PUT, and PATCH HTTP queries, as well as convert APIs.

Postman is built around a set of powerful technologies that are extremely easy to use.

Postman has become a useful tool for more than 8 million users. Postman is used for the following reasons:

1. Accessibility- After installing Postman on the device, simply login to the account to use it anywhere.
2. Use Collections- Users can create collections for their API calls using Postman. Each set can create multiple requests and subfolders. It will aid in the organisation of test suites.
3. Test development- Every API request should include verification of a successful HTTP response status to test checkpoints.
4. Automation Testing- The Collection Runner or Newman can conduct tests in multiple repetitions or iterations, saving time for repeated tests.
5. Creating Environments- The usage of numerous environments reduces test replication

because the same collection can be used in different settings.

6. Debugging- The postman console assists in successfully debugging the tests by tracking what data is being obtained.
7. Collaboration- To improve file sharing, you can import and export collections and environments. You can also share the collections via a direct connection.
8. Continuous integration- It can support continuous integration.

It's as simple as typing a URL into your browser to send a request. In Postman, we can quickly send requests to APIs. You can use an API request to access or deliver data from a data source.

An HTTP method is required to send the API request. POST, GET, DELETE, PUT, and PATCH are some of the most widely used methods.

→ GET: This HTTP method is used to access the data from an API.

→ POST: This method transmits new data.

→ DELETE: This is used to remove or delete the existing data.

→ PATCH: This method is used to update the existing data.

→ PUT: This method is used to update the existing data.

3.3 Typescript

TypeScript was invented and is maintained by Microsoft. It's a strict syntactic and semantic javascript framework with the feature of static typing as an option. It employs JavaScript and is designed for the construction of large apps. Because TypeScript is a javascript framework, existing JavaScript programs are also valid TypeScript programs.

Both server-side and client JavaScript applications can be built with TypeScript (as with

Node.js). You can use the TypeScript Analyzer or the Babel compiler to translate TypeScript to JavaScript.

TypeScript includes definition files that describe the structure of current JavaScript libraries, similar to how C++ header files describe the architecture of existing object files. Variables defined in the files can be used by other programs as if they were TypeScript properties with statically typed values. Third-party header files are included in popular frameworks like jQuery, Mongo, and D3.js. TypeScript annotations for Node.js fundamental modules are also available, allowing TypeScript to be used to write React.js apps.

The TypeScript compiler is written in TypeScript and then compiled to JavaScript. Its use is governed by the Apache License 2.0. TypeScript is supported as a first-class scripting language with C# or other Microsoft programs in Visual Studio 2013 Version 2 and later. Visual Studio 2012 now supports TypeScript thanks to an official plugin. The TypeScript was developed by Anders Hejlsberg, the principal architect of C# and developer of Delphi and Turbo Pascal.

3.4 MySQL:

MySQL is a Relational Database Management System (RDBMS) software that provides many things, which are as follows:

- It allows us to implement database operations on tables, rows, columns, and indexes.
- It defines the database relationship in the form of tables (collection of rows and columns), also known as relations.
- It provides the Referential Integrity between rows or columns of various tables.
- It allows us to update the table indexes automatically.

3.5 React

ReactJS is a Javascript framework for constructing declarative, fast, and adaptable user interface components. It's a component-based front-end framework that manages the app's view layer exclusively. Jordan Walke, a software engineer at Facebook, conceived the idea. Facebook invented and maintains it, and it's later used in Facebook products like WhatsApp and Instagram. Although ReactJS was designed around 2011 for Facebook's feed area, it was first publicly published in May 2013.

The most common website architecture today is MVC (model view controller). React is the view in the MVC design, whereas Flux or Redux is the architecture.

A ReactJS application is made up of numerous components, each being responsible for generating a reusable piece of HTML code. Components are the foundation of all React programs. Complex applications can be developed using simple building blocks by nesting these components with other components. ReactJS interconnects the DOM-based technique for populating data in the HTML DOM. Because the virtual DOM alters specific DOM components rather than refreshing the full DOM, it is fast.

To create a React app, we create React components that correspond to various aspects. The application structure is made up of these components, which are arranged into higher-level components. Consider a form containing the required fields, labels, and buttons, among other things. Each form component can be created as a React element, which we then combine to build the form component. The form elements would describe the structure and parts of the form.

ReactJS' major purpose is to construct a User Interface that aids program performance. It makes use of a virtual DOM (JavaScript objects), that speeds up the app. In JavaScript, the virtual DOM is quicker than the traditional DOM. ReactJS can also be used on server and client sides, as well as with other frameworks. Component and data structures are used to frame and make app maintenance easier.

ReactJS is quickly becoming the most popular JavaScript framework among web developers. It is an important part of the front-end ecology. The following are some of

ReactJS' key features:

- JSX

JavaScript XML is abbreviated as JSX. It's a syntax extension for JavaScript. ReactJS uses an XML or HTML-like syntax. This syntax is converted into React Framework JavaScript calls. It enhances ES6 to allow HTML-like text and JavaScript react code to coexist. It is not required to utilise JSX, however, it is strongly recommended in ReactJS.

- Components

Components are at the heart of ReactJS. A ReactJS application consists of several components, each with its logic and controls. These components are reusable, which makes it easier to keep the code clean while working on larger projects.

- One-way Data Binding

ReactJS is designed to handle one-way data binding or unidirectional data flow. One-way data-binding allows you to have more control over your application. If the data flow occurs oppositely, additional characteristics are required. Because components are designed to be immutable, and the data they contain cannot be modified, this is the case. Flux is a pattern that aids in data unidirectionality. This increases efficiency by making the application more versatile.

- Virtual DOM

The original DOM object is represented by a virtual DOM object. It functions similarly to a one-way data binding. The complete UI is re-rendered in virtual DOM representation whenever any changes are made to the web application. Then it compares the differences between the old and new DOM representations. After that, the true DOM will only update the things that have changed. This speeds up the application and eliminates memory waste.

- **Simplicity**

ReactJS makes use of a JSX file, which makes the program easy to code and comprehend. We already know that ReactJS is a component-based solution that allows you to reuse code as needed. This makes it easy to use and understand.

- **Performance**

The speed of ReactJS is well-known. This distinguishes it from other frameworks currently available. This is due to the fact that it manages a virtual DOM. The Document Object Model (DOM) is a cross-platform computer programming API for HTML, XML, and XHTML. The DOM is purely memory-based. As a result, we didn't even write directly to the DOM while creating a component. Instead, we'll create virtual components that will be converted into the DOM, resulting in smoother and faster performance.

3.6 Spring Boot

The term, “Microservices Framework” is often used when talking about Spring Boot. This is only partly true: Spring Boot is suitable for small services as well as for monolithic applications. These services or applications can be pure Web APIs or web applications, command-line programs, or orchestrating applications. In the end, Spring Boot orchestrates the Spring Framework. The Spring Framework provides application-level infrastructure ready, especially Dependency Injection and aspect-oriented programming. With these, building blocks become higher value concepts realised. Depending on the module, these may include transactions, security, and more. Spring Boot starts a Spring Container and configures - depending on the dependencies on the classpath.

Features of spring-boot

- Create stand-alone Spring applications
- Embed Tomcat, Jetty, or Undertow directly (no need to deploy WAR files)

- Provide opinionated 'starter' dependencies to simplify your build configuration
- Automatically configure Spring and 3rd party libraries whenever possible
- Provide production-ready features such as metrics, health checks, and externalised configuration
- Absolutely no code generation and no requirement for XML configuration

3.7 CI/CD

CI and CD stand for continuous integration and continuous delivery/continuous deployment. In very simple terms, CI is a modern software development practice in which incremental code changes are made frequently and reliably. Automated build-and-test steps triggered by CI ensure that code changes being merged into the repository are reliable. The code is then delivered quickly and seamlessly as a part of the CD process. In the software world, the CI/CD pipeline refers to the automation that enables incremental code changes from developers' desktops to be delivered quickly and reliably to production.

CI/CD allows organisations to ship software quickly and efficiently. CI/CD facilitates an effective process for getting products to market faster than ever before, continuously delivering code into production, and ensuring an ongoing flow of new features and bug fixes via the most efficient delivery method.

Continuous integration (CI) is practice that involves developers making small changes and checks to their code. Due to the scale of requirements and the number of steps involved, this process is automated to ensure that teams can build, test, and package their applications in a reliable and repeatable way. CI helps streamline code changes, thereby increasing time for developers to make changes and contribute to improved software.

Continuous delivery (CD) is the automated delivery of completed code to environments like testing and development. CD provides an automated and consistent way for code to be

delivered to these environments.

Continuous deployment is the next step of continuous delivery. Every change that passes the automated tests is automatically placed in production, resulting in many production deployments.

Continuous deployment should be the goal of most companies that are not constrained by regulatory or other requirements.

In short, CI is a set of practices performed *as developers are writing* code, and CD is a set of practices performed *after* the code is completed.

DevOps is a set of practices and tools designed to increase an organisation's ability to deliver applications and services faster than traditional software development processes. The increased speed of DevOps helps an organisation serve its customers more successfully and be more competitive in the market. In a DevOps environment, successful organisations “bake security in” to all phases of the development life cycle, a practice called *DevSecOps*.

The key practice of DevSecOps is integrating security into all DevOps workflows. By conducting security activities early and consistently throughout the software development life cycle (**SDLC**), organisations can ensure that they catch vulnerabilities as early as possible, and are better able to make informed decisions about risk and mitigation. In more traditional security practices, security is not addressed until the production stage, which is no longer compatible with the faster and more agile DevOps approach. Today, security tools must fit seamlessly into the developer workflow and the CI/CD pipeline in order to keep pace with DevOps and not slow development velocity.

The CI/CD pipeline is part of the broader DevOps/DevSecOps framework. In order to successfully implement and run a CI/CD pipeline, organisations need tools to prevent points of friction that slow down integration and delivery. Teams require an integrated toolchain of technologies to facilitate collaborative and unimpeded development efforts.

3.8 Figma

Figma is a cloud-based design tool that is similar to Sketch in functionality and features, but with big differences that make Figma better for team collaboration. For those sceptical of

such claims, we'll explain how Figma simplifies the design process and is more effective than other programs at helping designers and teams work together efficiently.

Figma works on any operating system that runs a web browser. Macs, Windows PCs, Linux machines, and even Chromebooks can be used with Figma. It is the only design tool of its type that does this, and in shops that use hardware running different operating systems, **everyone** can still share, open, and edit Figma files.

In many organisations, designers use Macs and developers use Windows PCs. Figma helps bring these groups together. Figma's universal nature also prevents the annoyance of PING-pong (where updated images are bounced back and forth between design team disciplines). In Figma, there is no need for a mediating mechanism to make design work available to everyone.

Because Figma is browser-based, teams can collaborate as they would in Google Docs. People viewing and editing a file are shown at the top of the app as circular avatars. Each person also has a named cursor, so tracking who is doing what is easy. Clicking on someone else's avatar zooms to what they are viewing at that time.

Real-time file collaboration helps mitigate "design drifting"—defined as either misinterpreting or straying from an agreed-upon design. Design drifting usually happens when an idea is conceived and quickly implemented while a project is in progress. Unfortunately, this often leads to deviating from the established design, causing friction and re-work.

Using Figma, a design lead can check in to see what the team is designing in real time by simply opening a shared file. If a designer somehow misinterprets the brief or user story, this feature allows the design lead to intervene, correct course, and save countless hours that would have otherwise been wasted. (By comparison, teams using Sketch have no immediate way of telling if designers are going astray.)

CHAPTER 4: STORYLINE OF THE PRODUCT

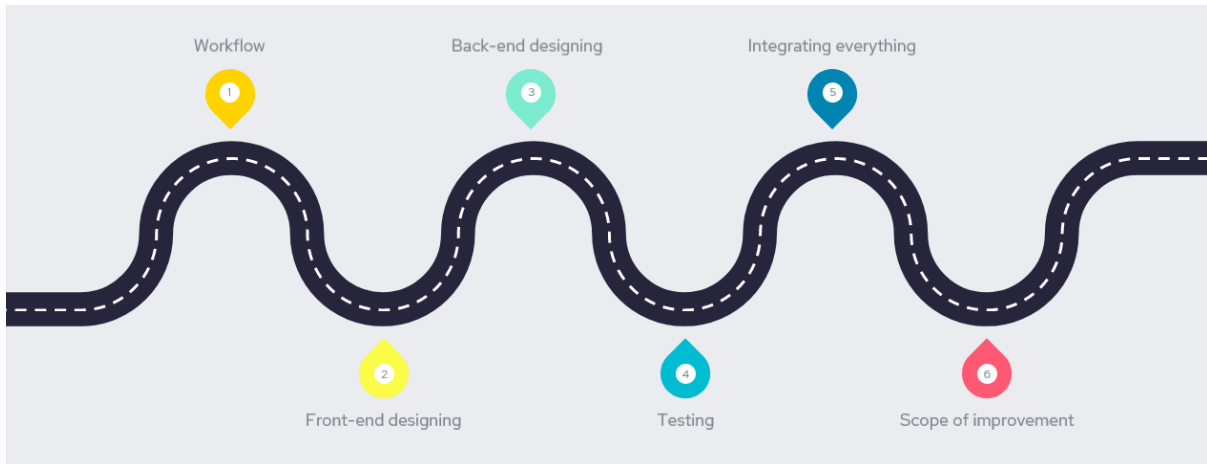


Fig. 2. Roadmap of the product

1. Figma Design: Designed the UI of login, signup and form pages for Save++ application.
2. Frontend Architecture (Design considerations) : Users can login using existing credentials and after proper authentication, signup using new credentials, use the user-friendly tax calculator that is divided in an elegant steps-wise manner for better user experience and offer supporting recommendations to help the user to enhance their tax saving experience.
3. Db Design: Database includes a number of tables that solve a different essential purpose. For this, a SQL database was employed. Some of the tables are User table that has basic personal details, UserDetails table that has income details and some general details and UserDeductions table that deals with all the fields under which users can claim their deductions.
4. Backend architecture : Design considerations: Architecture includes Entities like User, UserDetails, UserDeductions and entities necessary for managing JWT. Moreover there is a Controller that contains APIs for the signup page like adding a new user, login page like checking existing user details in database and authenticating the user, tax calculator page like calculating gross income, deductions and net taxable income. Then there are the services that handle the functioning of controller tasks and also, Repository, that takes care of object relational mapping with the database.

5. User Signup:

a. Frontend User Story:

- i. As a salaried employee, I should be able to register into the application so that I can login and start using the platform
- ii. Acceptance Criteria:
 1. The user has to provide a unique email id as his username
 2. Password should have at least a number, an uppercase letter and a special character
 3. Additional details like
 4. "FirstName", "LastName", "Username", "Email" , "Company Name" and "Password" have to be entered for new registration.

b. Backend User Story:

- i. As a salaried employee, I should be able to register into the application so that I can login and start using the platform
- ii. Acceptance Criteria:
 1. The user has to provide a unique email id as his username
 2. Password should have at least a number, an uppercase letter and a special character
 3. Additional details like
 4. "FirstName", "LastName", "Username", "Email" , "Company Name" and "Password" have to be entered for new registration.

6. User Login:

a. Frontend User Story:

- i. As an employee, I should be able to Login into the application so that I can add the necessary details to calculate my taxable income.
- ii. Pre-Condition:
- iii. The artist should have already completed the Signup
- iv. Acceptance Criteria:
 1. Only when the employee enters the correct email address and password, he/she is taken to the tax calculator page
 2. When the employee enters an incorrect email address or password, an error message "Incorrect Credentials" is displayed

3. When the employee clicks on “Forgot password?”, then the user is directed to the “Forgot Password?” screen
4. When the employee clicks on “Create new Account”, then the user is directed to the “Signup” page

b. Backend:

- i. Service POST /login
- ii. The service takes UID/ PWD as the request parameters and generates. JWT token with user details.
- iii. User should be able to login using – email and password
- iv. The auth token, user id should be returned in the response
- v. On incorrect details message an error message- “Either email or password is wrong” Is displayed
- vi. An api for updating the password should be created
- vii. All the error messages should be displayed in case the user enters incorrect details.
- viii. Unit test cases for all the scenarios should be covered

7. Tax Calculator:

a. Frontend: User Story:

- i. As a employee, I should be able to calculate tax by entering once successfully logged in
- ii. Acceptance Criteria:
 1. The user has to provide his/her unique username
 2. Password should have at least a number, an uppercase letter and a special character
- iii. Major components :
 1. Basic Information
 2. Additional details like Financial Year, Age and Gender have to be entered
 3. Income details
 4. The user has to provide details like basic income, business income, rental income and income from other sources

5. Deductions
6. Includes various options to claim deductions/exemptions from tax like Section 80C, Section 24, Section 10 and others
7. Summary
 - a. Includes the taxable income obtained under old and new income tax regimes
 - b. Shows gross income

- b. Backend: Service - GET /taxpage has to be created with request parameters as below

8. Tax Saving Suggestions:
 - a. Frontend:
 - i. User is given suggestions/recommendations on the summary step of the tax calculator page.
 - ii. The optimal choice of regime based on the taxable income obtained under old and new tax regime is recommended to the user
 - iii. The top tax deduction options are suggested to the user to help them save more tax.

 - b. Backend:
 - i. Service - POST /user/register has to be created with request parameters as below
 - ii. Error messages should be displayed in case the user enters incorrect details.
 - iii. Unit test cases for all the scenarios should be covered

9. User Dashboard: User Story:
 - a. Users can upload their image, update their description, firstname, lastname, age, gender and other details if required.
 - b. Recent history of tax calculations displayed.
 - c. Infographic features to display the gross income and tax saved out of it.

CHAPTER 5: PERFORMANCE ANALYSIS

4.1. Testing

Testing may not matter much when you are working on a small app but it becomes very important in large scale apps which are worked on by many developers.

Even a small change has the potential to break something in your app and if we have tests written for different features of our app, we can easily detect when a change breaks something in our app - before we ship our code to production.

React Testing Library is an alternative to Enzyme. While Enzyme lets us test the implementation details of React components, React Testing Library helps us test the behaviour of our React components from the perspective of the users that will use our app.

React Testing Library is used with a testing framework. Although React Testing Library is not dependent on any specific testing framework and can be used with any framework, the official docs of this library recommend using the library with Jest. In this tutorial, we will use Jest as our testing framework.

Testing is an important part of software development. You can always test your project manually by building it and then running it. But manual testing implies that there's a human involved and thus the potential for human error and inconsistent testing. Automated tests are more consistent and repeatable.

Recognizing this, the Spring Initialize gives you a test class to get started. The following listing shows the baseline test class.

There's not much to be seen in `StudentmanagementApplicationTests`: the one test method in the class is empty. Even so, this test class does perform an essential check to ensure that the Spring application context can be loaded successfully. If you make any changes that prevent the Spring application context from being created, this test fails, and you can react by fixing the problem.

The `@SpringBootTest` annotation tells JUnit to bootstrap the test with Spring Boot capabilities. Just like `@SpringBootApplication`, `@SpringBootTest` is a composite annotation,

which is itself annotated with `@ExtendWith (SpringExtension.class)`, to add Spring testing capabilities to JUnit 5. For now, though, it's enough to think of this as the test class equivalent of calling `SpringApplication.run()` in a `main ()` method.

Finally, there's the test method itself. Although `@SpringBootTest` is tasked with loading the Spring application context for the test, it won't have anything to do if there aren't any test methods. Even without any assertions or code of any kind, this empty test method will prompt the two annotations to do their job and load the Spring application context. If there are any problems in doing so, the test fails.

4.2 Best Practices

- Used standard naming conventions for naming files, packages and methods.
- Created configuration files to hold environment variables and service component integration.
- Created configuration files to hold environment variables and service component integration.
- Created Junit Test cases for all services and controllers
- Implemented REST API Documentations for all services using Swagger
- CI/CD pipelines set on master branch for automatic deployment
- Setup CI/CD on "master" branch
- All the development and merging of features is done on separate branch.
- For each feature, a new branch is checked out and later merged with the branch for development, and on successful integration it is merged with the "main" for deployment
- Commits are made on regular intervals and meaningful commit changes are used.

4.2 Results

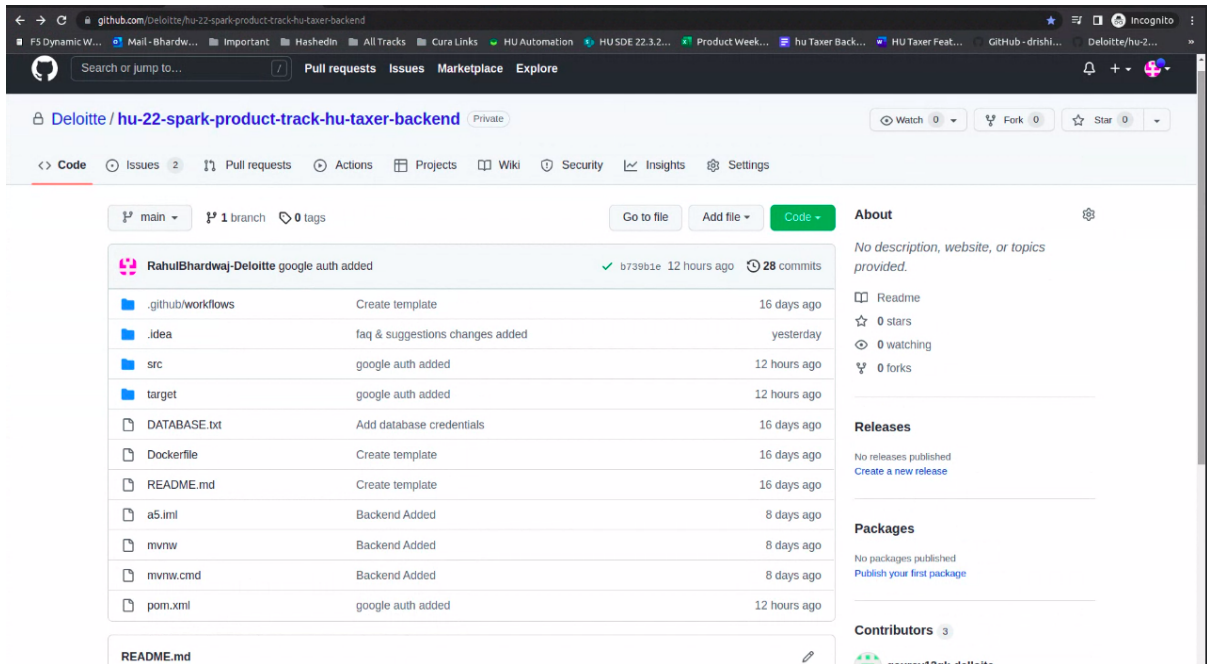


Fig. Backend Repository

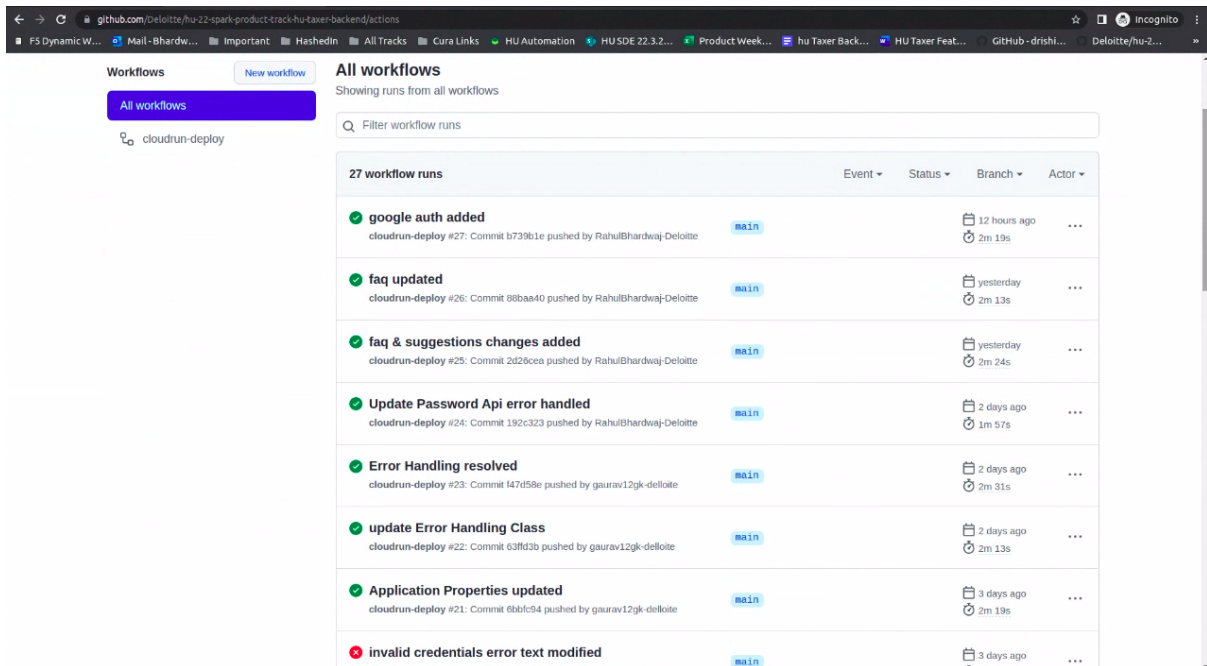


Fig. Backend Deployment Workflows

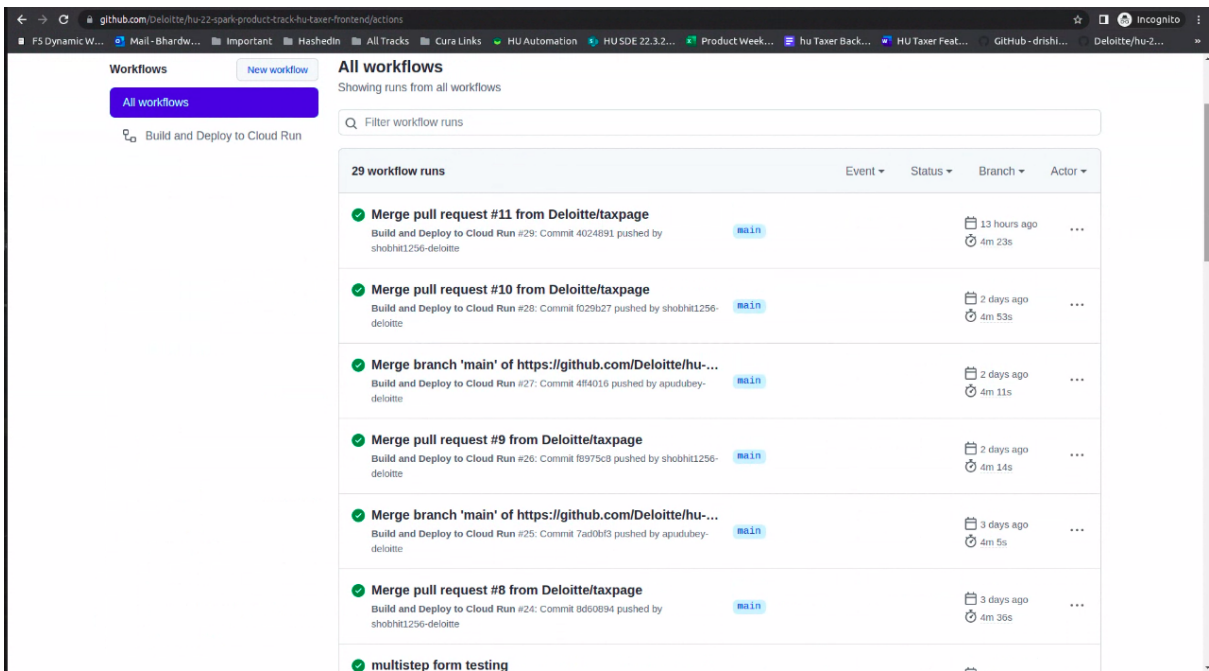


Fig. Frontend Deployment Records

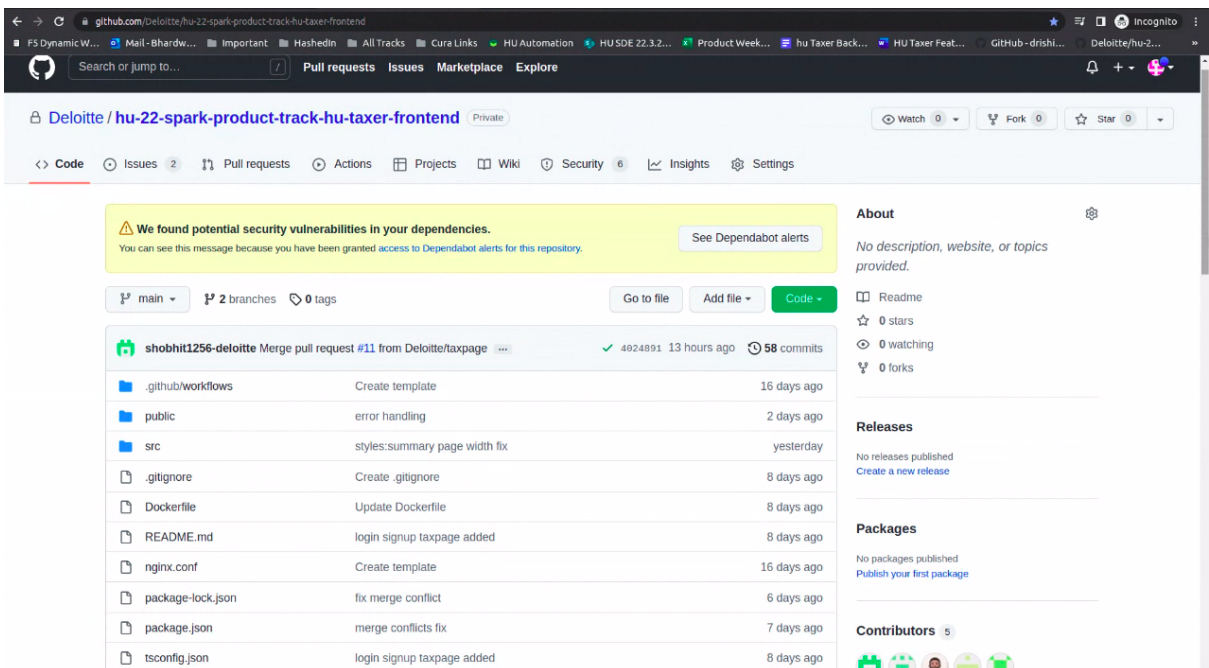


Fig. Frontend Repository

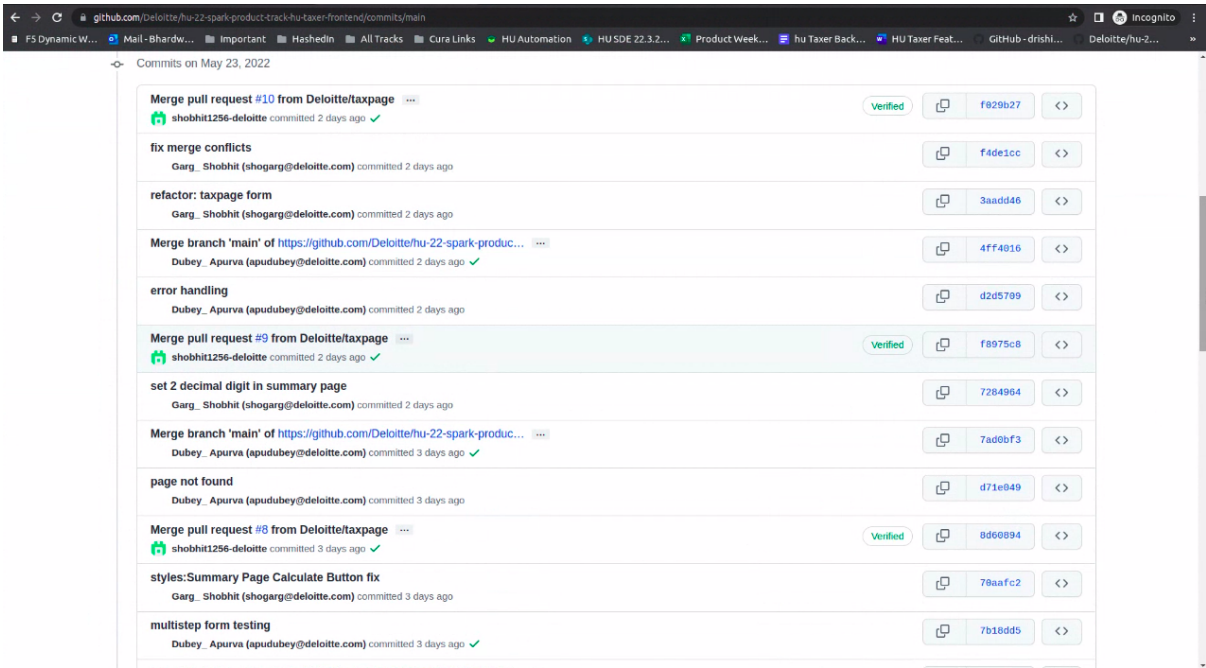


Fig. Commits History

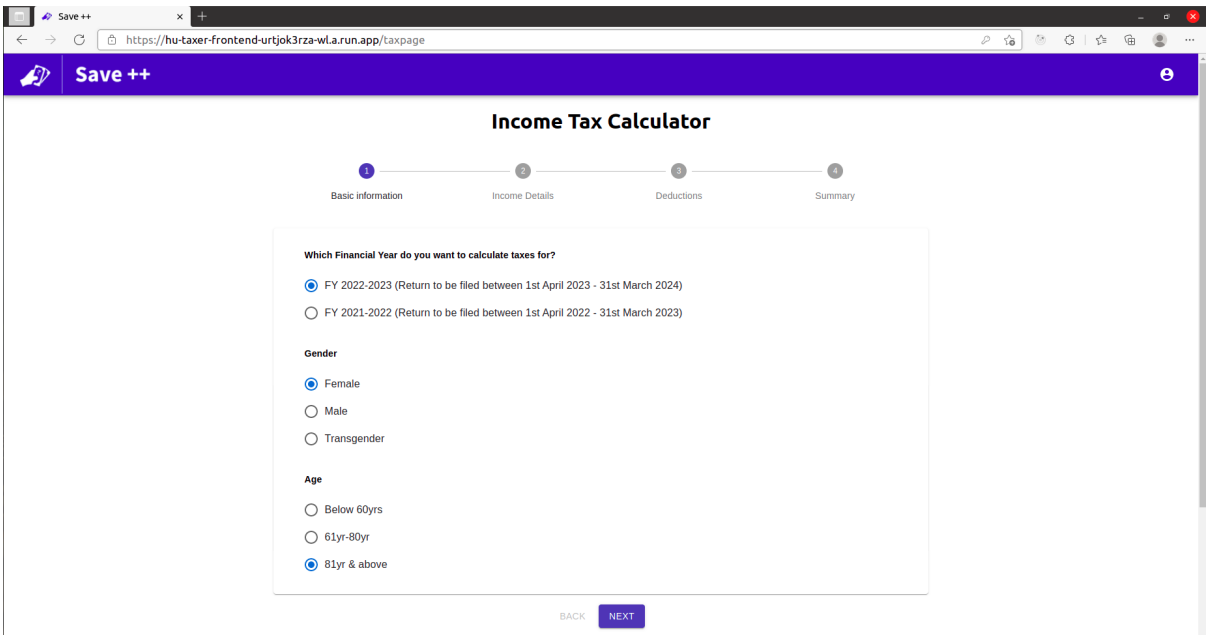


Fig. Income tax Calculator form

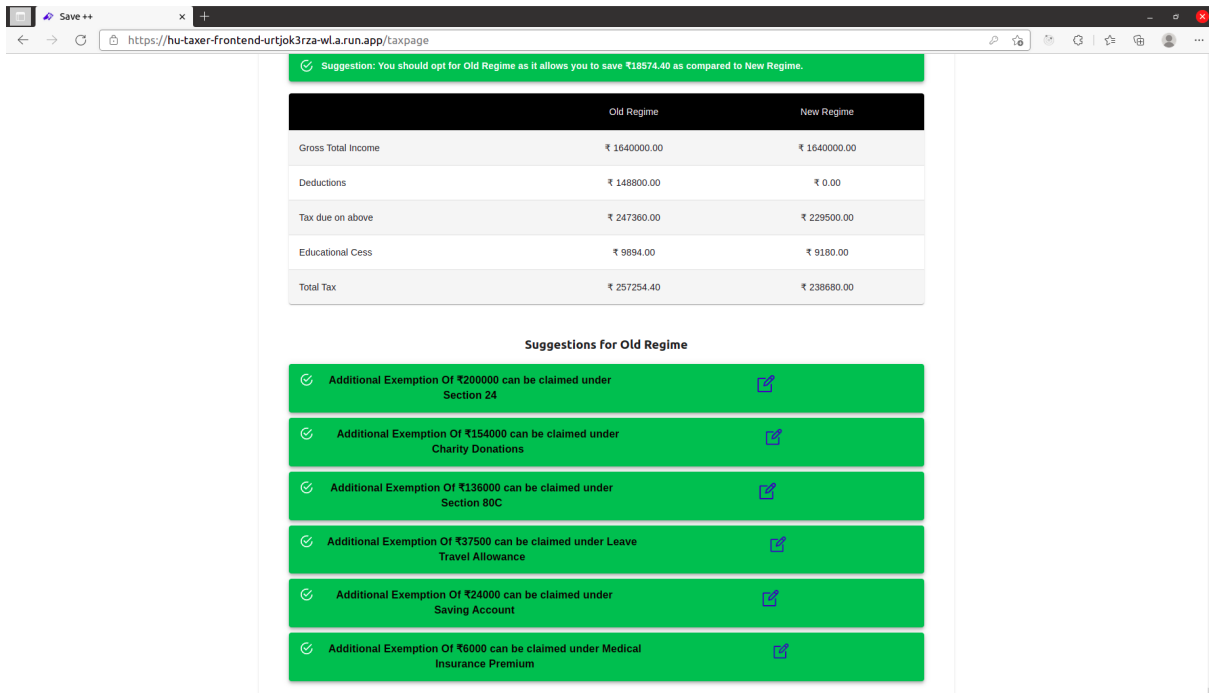


Fig. Tax Report

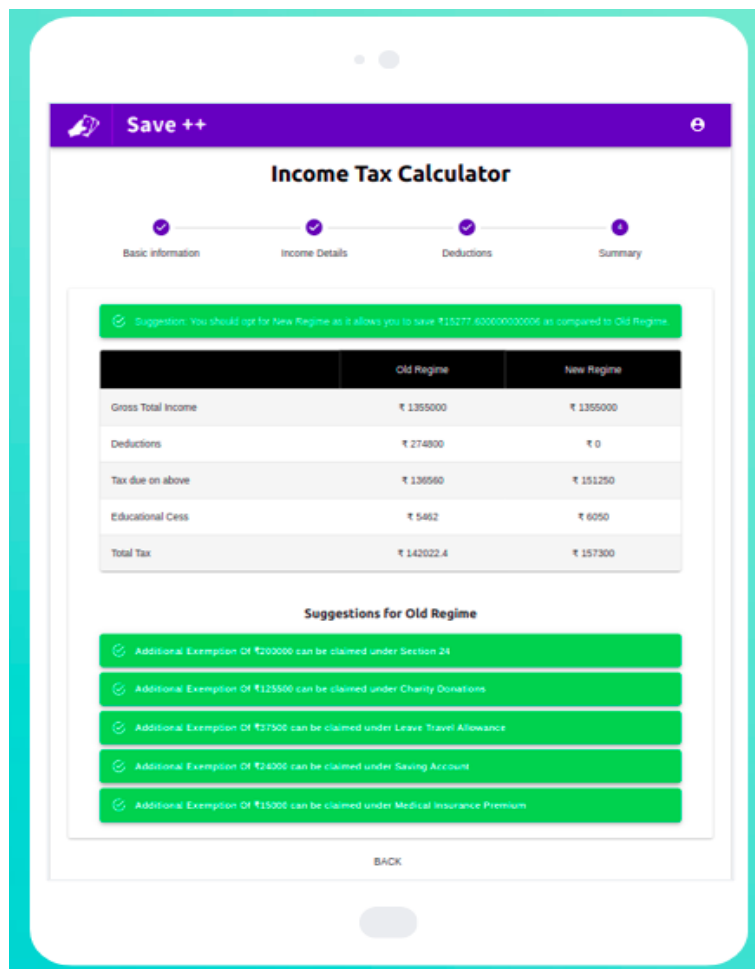


Fig. Tablet View Of Application

Under Section 10

Child Education allowance ⓘ 13000	Child Hostel Allowance ⓘ 15000
--	---

HRA

Actual House Rent Allowance received ⓘ 0	Actual Rent paid ⓘ 0
---	---

Do you live in Metro City (Mumbai, Chennai, Kolkata, Delhi)

True False

Leave Travel Allowance ⓘ

Amount ⓘ

12500

Under Section 24 ⓘ

Rent of Property ⓘ 0	Property Tax ⓘ 0
---	---

Interest on Home loan taken

Other Deductions

Medical insurance premium (s/y 80D) ⓘ 10000	Additional contribution towards NPS (s/y 80CCD(1B)) ⓘ 0
Employer's contribution toward EPF ⓘ 0	Donations (s/y 80G) ⓘ 10000
Employer's contribution toward NPS (s/y 80CCD) ⓘ 12400	Interest on loan for higher education (s/y 80E) ⓘ 900
Interest on deposits in saving account (s/y 80TTA) ⓘ 1000	
Interest on loan taken for Residential House (s/y 80EE) ⓘ 0	

Fig. Mobile View

CHAPTER 6: CONCLUSION

5.1 Conclusions

In conclusion, this internship has become a wonderful and rewarding experience. I can say that my stay with Hashedin By Deloitte was quite beneficial to me. Needless to say, the technical components of my work aren't perfect and, given enough time, could be improved. As someone who had no prior experience with full stack development, I think the effort I spent learning and understanding about was well spent, as it helped to construct a fully effective web service. Two of the most significant lessons I've learned are time management and self-motivation.

5.2 Future Work

- Dynamic React Js Forms
- A landing page to describe more about the application with infographics
- Cover other categories of taxpayers

REFERENCES

1. Kavya Guntupally, Kyle Dumas, Giri Prakash, Ranjeet Devarakonda, Wade Darnell, Maggie Davis, Richard Cederwall, "Spring Boot based REST API to Improve Data Quality Report Generation for Big Scientific Data: ARM Data Center Example", *Earth Science Informatics*, vol. 14, pp. 1487, 2021. <https://ieeexplore.ieee.org/document/8621924>
2. Java Brains
3. Spring Boot Book, Spring in Action
<https://books.google.co.in/books?hl=en&lr=&id=2zVbEAAAQBAJ&oi=fnd&pg=PA1&dq=%22spring+boot%22++research+paper&ots=PgTGCr3yMj&sig=KxInZhOadUGRh63fCOhVmLAVtZ4#v=onepage&q&f=true>
4. Income Tax Government Website,
<https://www.incometaxindia.gov.in/pages/tools/income-tax-calculator.aspx>
5. Swagger Documentation, <https://swagger.io/docs/>
6. React Testing Documentation, <https://testing-library.com/docs/react-testing-library/intro/>
7. Spring Boot Testing Documentation,
<https://docs.spring.io/spring-boot/docs/1.5.2.RELEASE/reference/html/boot-features-testing.html>
8. JWT Authentication Documentation, <https://javatodev.com/spring-boot-jwt-authentication/>