Department of Computer Science
University of Delhi

Internship Project Report

Regulatory Reporting Solutions "RAPTOR"

- Sajal Maheshwari (21234747040)

Declaration

Internship done under the guidance of **Dr. Kuldeep Singh** and mentorship under **Mr. Sunil Meena**

The work presented in this project entitled **"Regulatory Reporting Solutions (RAPTOR)"** presented in partial fulfillment for the award of the degree of **Master of Science (Computer Science)** submitted in the **Department of Computer Science, University of Delhi, Delhi** is an authentic work carried out from **9 Jan, 2023 to 9 JULY, 2023** under the guidance of **Dr. Kuldeep Singh, University of Delhi**.

——

Indus Valley Partners

Company

Indus Valley Partners is a tech based consulting firm focused on the capital market domain since 2000.

With its three main segments of business- IVP Products, IVP Managed Services and IVP Consultancy, the company helps its clients stay a step ahead in the capital market domain.

CEO and Founder

Gurvinder Singh

Expertise: New Fund Launches, Hedge Fund Trading Strategies, Fund Administration

CCO

Gaurav Aggarwal

Expertise: : Portfolio Accounting, Performance Attribution, Regulatory Reporting

Products under Indus Valley Partners

Company's products and services are divided into two main segments -

Front Office Products
- Enterprise Data Management
- Portfolio Solution
- Security and Reference Master
- Decision Science
- Data Warehouse
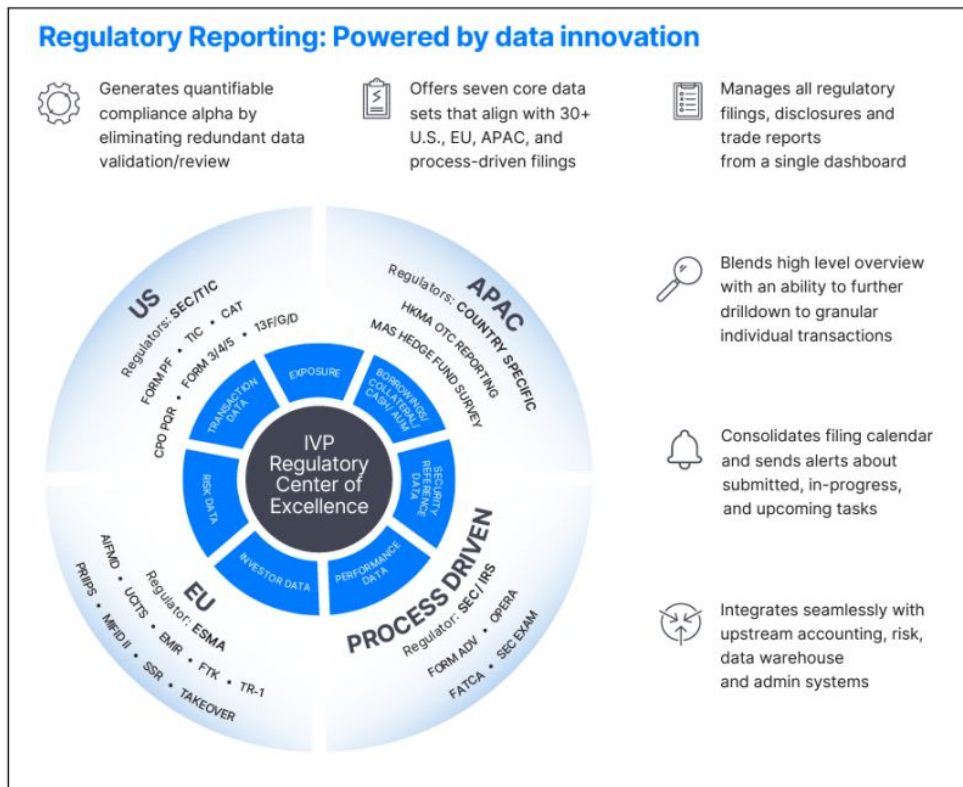- Order Management System

Middle/Back Office Products
- Reconciliation Solution
- Cash Management Solution
- ALT Data Analyzer
- **Regulatory Reporting Solution**
- Expense Allocation Solution
- ESG Management
- Treasury Management
- NAV Solution
- Pricing & Valuation Automation Solution

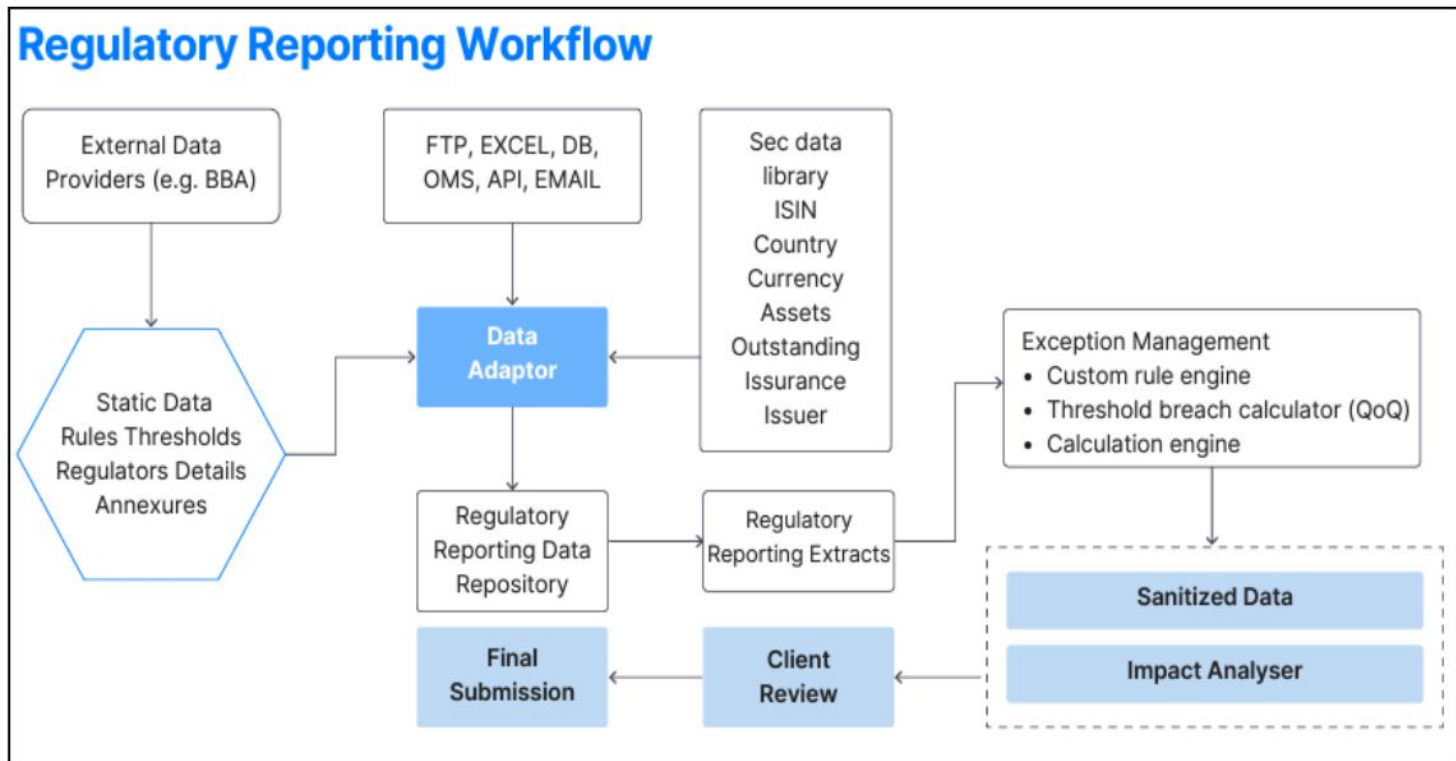*For more details about these products, please visit the company website*

Regulatory Reporting Solution "Raptor"

Regulatory Reporting Solutions Overview

Regulatory Reporting brings automation and data expertise to the regulatory technology ecosystem, offering seven core data sets that can drive more than 30 global filings with a single, streamlined workflow. Together with the company's digital-first managed services, Raptor uses a cloud-based deployment model to bring enormous economies of scale to compliance reporting.


Regulatory Reporting: Powered by data innovation

Generates quantifiable compliance alpha by eliminating redundant data validation/review

Offers seven core data sets that align with 30+ U.S., EU, APAC, and process-driven filings

Manages all regulatory filings, disclosures and trade reports from a single dashboard

Blends high level overview with an ability to further drilldown to granular individual transactions

Consolidates filing calendar and sends alerts about submitted, in-progress, and upcoming tasks

Integrates seamlessly with upstream accounting, risk, data warehouse and admin systems

Regulatory Reporting Solutions Workflow

Regulatory Reporting Workflow

- External Data Providers (e.g. BBA)
- FTP, EXCEL, DB, OMS, API, EMAIL
- Sec data library ISIN Country Currency Assets Outstanding Issurance Issuer
- Static Data Rules Thresholds Regulators Details Annexures
- Data Adaptor
- Exception Management
  - Custom rule engine
  - Threshold breach calculator (QoQ)
  - Calculation engine
- Regulatory Reporting Data Repository
- Regulatory Reporting Extracts
- Sanitized Data
- Impact Analyser
- Final Submission
- Client Review

Technologies in Raptor

React 18, Typescript, Redux Store, Redux Saga, Jest

Storing all the data, fillings, regulator's validations etc.

Core of Raptor, responsible for building dynamic components with the help of JQuery and LINQ.

| React | C# Web Services | SQL | ASP.Net | JS, JQuery & LINQ |
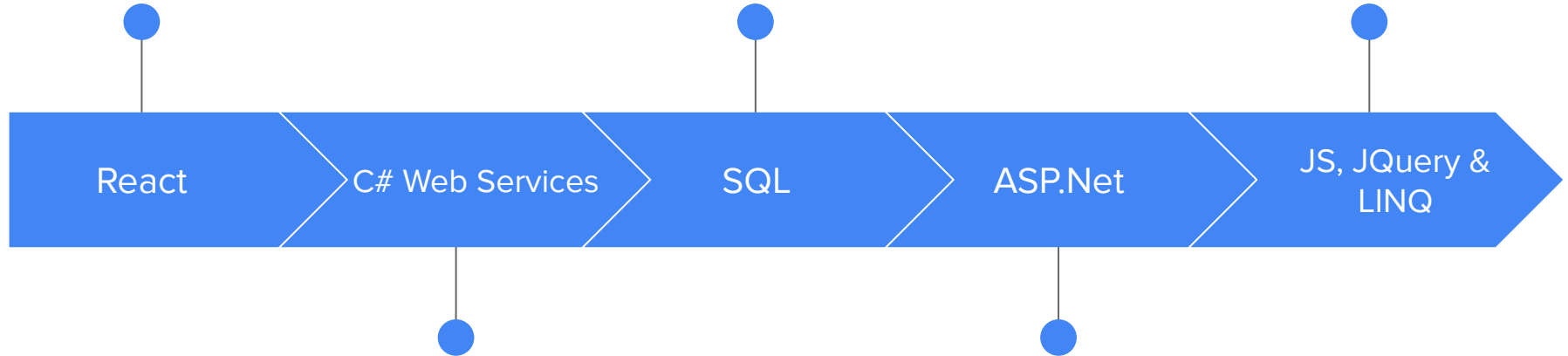
Develop Web Services

Modern framework accessed the web services, generate the dynamic HTML based code using javascript.

Project Work

Overview

The project work consists of most of the development in the **Frontend - React(Typescript), Redux, CSS Styling and Javascript**. Backend web services are developed, if required for these task, using the C# and LINQ.

The project work is divided into **three tasks**
- **Question Review Component**
- **Question Preview Component**
- **Question Footer Component**

Question Review Component

The design and requirements were taken from the existing Raptor Application. Screen captures are as follows -

Question Review Component

Development Hierarchy of the code is:
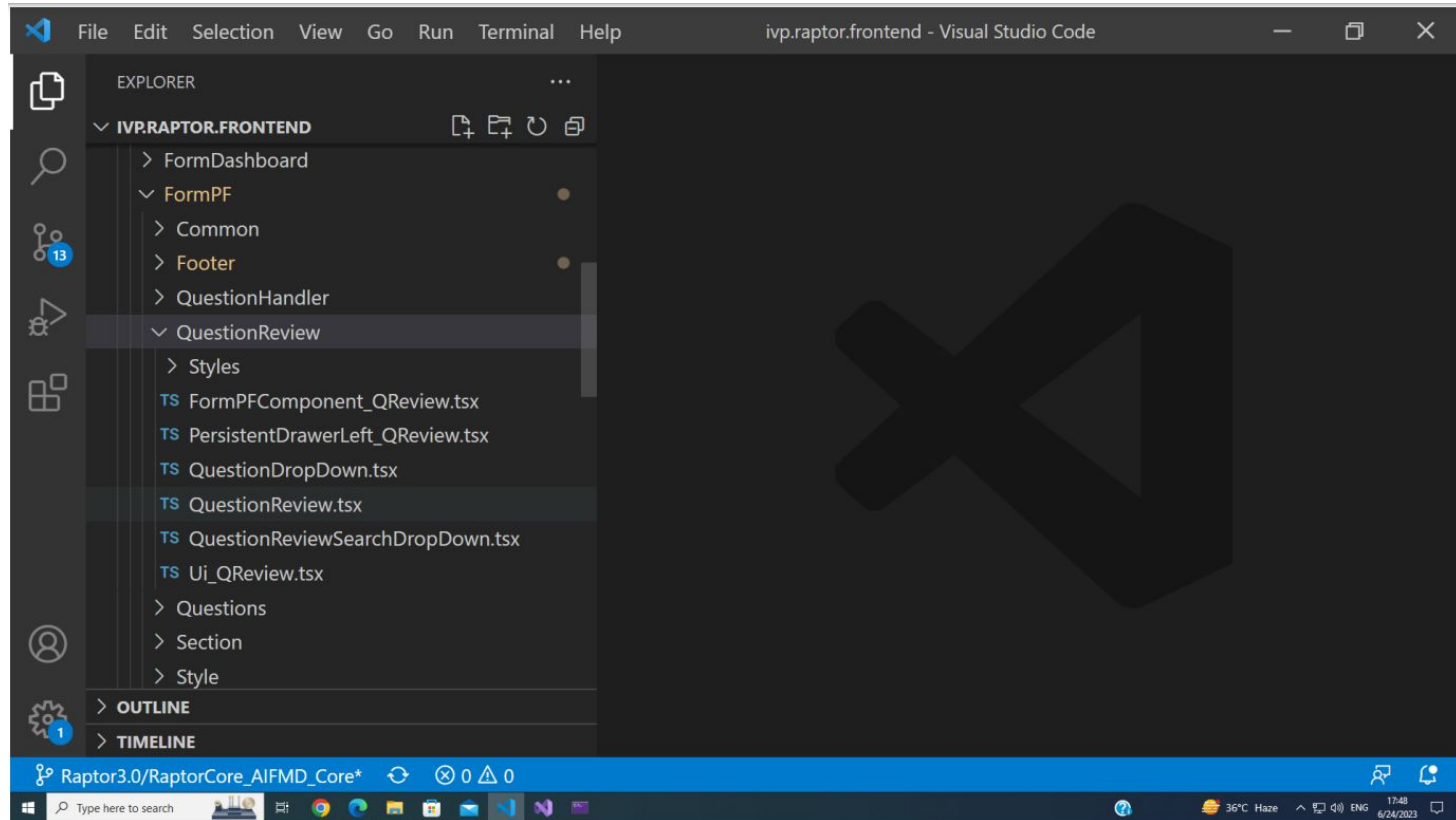
Question Review Component

FormPFComponent_QReview.tsx

```
return (
        <div id="pageContainer" className={cssClasses.FormPFComponent_pageTabPageContainer}>
                <Box className={cssClasses.FormPFComponent_OuterBox}>
                        <Box className={cssClasses.FormPFComponent_InnerBox}>
                                <div className={cssClasses.FormPFComponent_OuterDiv}>
                                        <div className={cssClasses.FormPFComponent_InnerDiv}>
                                                <Link className={cssClasses.FormPFComponent_pageTabBreadCrumpDisabled} to="/">
                                                        <divclassName={cssClasses.FormPFComponent_pageTabBreadCrumpDisabled}>Summary Dashboard</div>
                                                </Link>
                                                <div className={cssClasses.FormPFComponent_pageTabBreadCrumpArrow}>&gt;&gt;</div>
                                                <Link className={cssClasses.FormPFComponent_pageTabBreadCrumpDisabled} to={formSummaryUrl}>
                                                        <div className={cssClasses.FormPFComponent_pageTabBreadCrumpDisabled}> Form Dashboard </div>
                                                </Link>
                                                <div className={cssClasses.FormPFComponent_pageTabBreadCrumpArrow}>&gt;&gt;</div>
                                                <div className={cssClasses.FormPFComponent_pageTabBreadCrump}>Form</div>
                                        </div>
                                </div>
                                <div className={cssClasses.FormPFComponent_TabDiv}>
                                        <lvpTabControl onChange={tabChangeHandler} tabsObj={tabsObj} variant='lvpTabControlSecondaryStandard' controlId="1"
                                                tabStyling={{fontSize: "0.813rem"}} />
                                </div>
                        </Box>
                        {tab == "Form" && <Ui_QReview refresh={refresh} setRefresh={setRefresh} tab={tab} formId={formId}
                        fundId={fundId} questionId={questionId} initialView={initialView} previewType="question" />}
                        {tab == "Fund" && <Ui_QReview refresh={refresh} setRefresh={setRefresh} tab={tab} formId={formId}
                        fundId={fundId} questionId={questionId} initialView={initialView} previewType="question" />}
                </Box>
        </div>
)
```

Question Review Component

Ui_QReview.tsx

```
return (
        FormPFreviewJSONdataLoading ?<ThreeDotSpinner/>:<div id="FormPFUi_QuestionReview">
                <Grid container spacing={2}>
                        <Grid item xs={open ? 2.2 : 0.2} sm={open ? 2.2 : 0.2} md={open ? 2.2 : 0.2}>
                                <div>
                                        <PersistentDrawerLeft_QReview initialView={props.initialView} previewType={props.previewType} fundId={props.fundId}
                                                setFundSelected = {setfundSelected} questionSelected={questionSelected}
                                                FormProperties={FormProperties} data={props.tab == "Fund" ? LeftMenuQuestionData: undefined}
                                                open={open} setOpen={setOpen} />
                                </div>
                        </Grid>
                        <Grid item xs={open ? 12 - 2.2 : 12 - 0.2} sm={open ? 12 - 2.2 : 12 - 0.2} md={open ? 12 - 2.2 : 12 - 0.2}>
                                <div>
                                        <QuestionReviewDropDown initialView = {props.initialView} printState={ printState} setPrintState={setPrintState} tab={props.tab}
                                                refresh={props.refresh} setRefresh={props.setRefresh} FormProperties={FormProperties}
                                                previewType={props.previewType} formDetailsQuestionReview={FormPFformDetailsQuestionReview}
                                                questionSelected={questionSelected} setSelectedQuestion={getQuestion} setQuestion={setQuestion}
                                                questionData={questionData} />
                                </div>
                                <div id="questionContainer">
                                        <div data-testid="FormPFUiquestionContainer" id="questionContainerInner" className={open == false
                                                ? cssClasses.FormPFUI_navBodyClose : cssClasses.FormPFUI_navBodyOpen}>
                                                < QuestionFilter      formId = {FormProperties.formId} ref={componentRef} questionData={questionData}
                                                        setQuestion={setQuestion} FormProperties={FormProperties} sectionSelected={questionSelected}
                                                        tab={props.tab} previewType={props.previewType} />
                                        </div>
                                </div>
                        </Grid>
                </Grid>
        </div>
)
```

Question Review Component

Preview of the Question UI Component is as follows:

Question Review Component

Preview of the Question Review component is as follows:

Task 2

Question Preview Component

Problem Statement

Using the existing code, create the logic and develop the Question Preview page for the Raptor.

Use the requirements collected during the task "Debug based tasks for the requirement collection for the project work" to complete the task.

Estimated Delivery: 7 working days

Question Preview Component

Question Preview Component enable its user to walkthrough the form application without disturbing its value filling informations or changing any form configuration throughout the walkthrough period, as the Question Preview disables all the editable fields throughout the application and also hides all the question ui header/footer and bulk actions utilities from the user.

Since the component is reusing the same code, multiple checks (if...else blocks and ternary operations) throughout the react application code have been set according to the requirements specified.

Question Preview Component

```
return ( <>{isRender() && <ThemeProvider theme={theme}><div id="pageContainer" className={cssClasses.FormPFComponent_pageTabPageContainer}>
                        <Box className={cssClasses.FormPFComponent_OuterBox}>
                            <Box className={cssClasses.FormPFComponent_InnerBox}>
                                <div className={cssClasses.FormPFComponent_OuterDiv}><div className={cssClasses.FormPFComponent_InnerDiv}>
                                    <CustomBreadcrumbs breadcrumbs={breadcrumbs} /></div>
                                </div>
                                <div className={cssClasses.FormPFComponent_TabDiv}>
                                    <lvpTabControl    onChange={tabChangeHandler}    tabsObj={tabsObj}    variant='lvpTabControlSecondaryStandard'
                                    controlId="1"
                                            tabStyling={{fontSize:"0.813rem"}} />
                                </div>
                                <div className={cssClasses.FormPFComponent_OuterFormConfigDiv}>
                                    <div className={cssClasses.FormPFComponent_InnerFormConfigDiv}>
                                        {formconfigpermission?.[0]?.IsConfig &&
                                        <Link className={cssClasses.FormPFComponent_pageTabBreadCrumpDisabled}  to={formConfigUrl} >
                                        <i className={`fa fa-cog fa-2 ${cssClasses.FormPFComponent_pageTabFormConfigImage}`}></i></Link>}
                                           
                                        <Link className={cssClasses.FormPFComponent_pageTabBreadCrumpDisabled} to={formTemplateUrl} >
                                                <img alt="Template Upload Menu"
                                                 className={cssClasses.FormPFComponent_pageTabTemplateUploadIcon}
                                                src={String(TemplateUploadButton)} />
                                        </Link>
                                           
                                        <CalendarMonthIcon className={cssClasses.FormPFComponent_pageTabCalenderIcon} />
                                        <div className={cssClasses.FormPFComponent_pageTabFormdate}>{formpfdate}</div>
                                    </div>
                                </div>
                            </Box>
                    {tab == "Form" && <FormPFUI tab={tab} formId={formId} sectionId={sectionId} fundId={fundId} questionId={questionId}
                                pageType={props?.pageType} previewInitialState={previewInitialState} />}
                    {tab == "Fund" && <FormPFUI tab={tab} formId={formId} sectionId={sectionId} fundId={fundId} questionId={questionId}
                                pageType={props?.pageType} previewInitialState={previewInitialState} />}
            </Box></div></ThemeProvider>}</>);
```

Question Review Component

Preview of the Question UI Component is as follows:

Question Review Component

Preview of the Question Preview Component is as follows:

Task 3

Question Footer Component

Problem Statement

Develop all the Question Footer Utilities for the Raptor.

Use the requirements collected during the task "Debug based tasks for the requirement collection for the project work" to complete the task.

Estimated Delivery: 20 working days

Question Footer Component

Question Footer is a set of utilities that enable its user to perform certain tasks on a particular question in the form filling. These operations are -

- Comment and Attachments
- Calculation Tree
- QoQ(Quarter over Quarter)
- User Permissions
- Related Questions
- Validation Alerts
- FAQs
- Glossary
- Audit

Question Footer Component

Development Hierarchy of the code is:
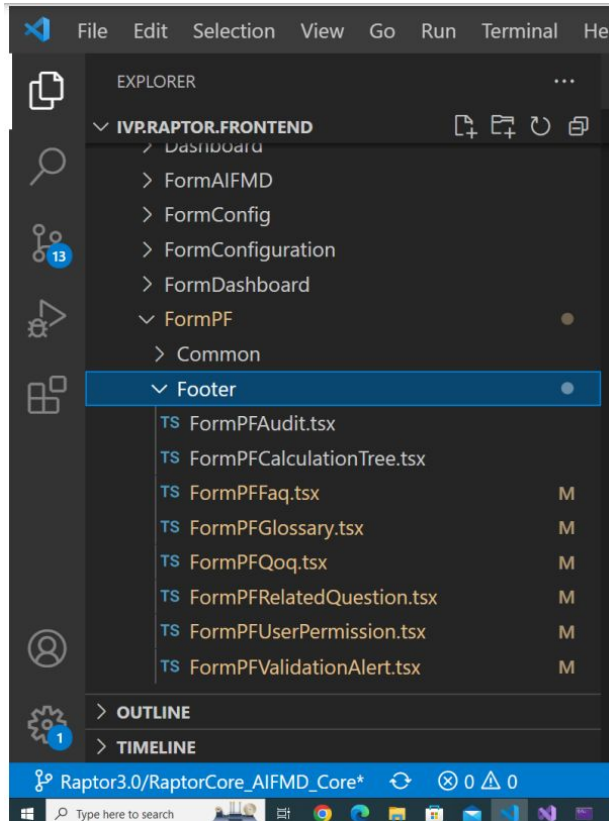
FormPFAudit.tsx
```
return (
    <div id={props.FormProperties.formId + "_" +
        props.FormProperties.currentFundId + "_" +
        props.questionId + "_" + "support"}
        className={cssClasses.Audit_questionSupport}>
        <div id={props.FormProperties.formId + "_" +
            props.FormProperties.currentFundId + "_" +
            props.questionId + "_support_iframe_Audit"}
            className={cssClasses.Audit_ActiveArrow} style={{
                'left': props.leftStyle }}>
        </div>
        <div className={cssClasses.Audit_outerdiv}>
            <div className={cssClasses.Audit_header}>
            <div className={cssClasses.Audit_commentheading}>
            AUDIT
            </div>
            <div className={cssClasses.Audit_tab}>
                <IvpTabControl onChange={tabChangeHandler}
                tabsObj={tabsObj} controlId="1"
                variant='IvpTabControlSecondaryStandard'
                tabStyling={{fontSize : "0.813rem"}} />
            </div>
            </div>
        {GetAuditWrapper(AuditData, ApprovalAuditData,
        ManualAuditData,ApprovalAuditDataSnapshot,IsAuditSnapshot)}
        </div>
    </div>
);
```

Question Footer Component

GetAuditWrapper method for FormPFAudit.tsx

```
const GetAuditWrapper = (AuditData: any, ApprovalAuditData: any, ManualAuditData: any,ApprovalAuditDataSnapshot:any,IsAuditSnapshot:boolean) => {
        return (<>{ tab == "Question Level Audit" &&<div className={cssClasses.Audit_grid}>
                <AdaptableGrid    ref={GridRef} adaptableOptions={AuditData["adaptableOptions"]}
                                  GridProp={AuditData["gridProp"]} gridOptions={AuditData["gridOptions"]} />
                </div>}
                { tab == "Approval Audit" && <div className={cssClasses.Audit_grid}>{IsAuditSnapshot ?<>
                                            <i className={`fa fa-times ${cssClasses.Audit_gridClose}`} aria-hidden="true" onClick={()=>setIsAuditSnapshot(false)}></i>
                <AdaptableGrid    ref={GridRef} adaptableOptions={ApprovalAuditDataSnapshot["adaptableOptions"]}
                                  GridProp={ApprovalAuditDataSnapshot["gridProp"]} gridOptions={ApprovalAuditDataSnapshot["gridOptions"]} />
                </>:
                <AdaptableGrid    ref={GridRef} adaptableOptions={ApprovalAuditData["adaptableOptions"]}
                                  GridProp={ApprovalAuditData["gridProp"]} gridOptions={ApprovalAuditData["gridOptions"]} />}
                </div>}
                {tab == "Manual Override Audit" && <div className={cssClasses.Audit_grid}>
                <AdaptableGrid    ref={GridRef} adaptableOptions={ManualAuditData["adaptableOptions"]} GridProp={ManualAuditData["gridProp"]}
                                  gridOptions={ManualAuditData["gridOptions"]} />
                </div>
                }
                </>);
}
```

Question Footer Component

Screen Snips of the UI for the utilities are as follows-
FormPFQoq.tsx

Question Footer Component

Screen Snips of the UI for the utilities are as follows-
FormPFUserPermissions.tsx

Question Footer Component

Screen Snips of the UI for the utilities are as follows-
FormPFValidationAlert.tsx

Question Footer Component

Screen Snips of the UI for the utilities are as follows-
FormPFAudit.tsx

Results

Result 1

Efforts taken for each component

Task Name	Estimated Efforts	Actual Efforts
Question Review Component	10 days	12 days
Question Preview Component	7 days	6 days
Question Footer Component	20 days	18 days

Result 2

Completion of component

Task Name	Development Completed
Question Review Component	Yes
Question Preview Component	Yes
Question Footer Component	Yes

Result 3

Submit for the production

Task Name	Sent to Deployment
Question Review Component	Under testing
Question Preview Component	Yes
Question Footer Component	Yes

Conclusion

- Added skills
- Advantage of working in the team
- Experience the rich culture of healthy office suite

Future Scope

- Work on different forms like Form AIFMD, CPO, H16 and 50 others
- Work on the client side, resolving their issues within promised time period
- Train other team members who will switch from the ASP.NET to React
- Build the project training schedule for the upcoming freshers

Thank You!