TITLE OF THE PROJECT

Scholar Tracker

A Robust Web Platform for Research Scholar & Academic Institution

OBJECTIVE AND SCOPE

OBJECTIVE:

The objective of Scholar Tracker is to develop a web-based application to help academic institutions efficiently track and manage the progress of scholars. The system will simplify the process of monitoring academic milestones, managing research outputs, and enhancing communication between scholars, supervisors, and administrative and evaluation committee. It aims to ensure timely progress, improved organization, and transparency in academic activities.

SCOPE:

1. Scholar Profile Management

Maintain detailed profiles for scholars, including personal details, academic records, and research topics. Centralize all scholar-related information for easy access.

2. Progress Tracking

Track key milestones such as coursework completion, rdc's tracking, thesis progress, and submission deadlines. Provide automated notifications and reminders for important tasks.

3. Document Management

Facilitate the upload, storage, and review of important documents like research papers, reports, and publications. Enable supervisors to review and approve documents within the system.

4. Communication and Interaction

Provide tools for scheduling meetings, sharing feedback to facilitating communication between scholars and supervisors.

5. Reporting and Analytics

Generate performance reports and visual progress charts for scholars and supervisors. Allow administrative committee to oversee the overall progress of multiple scholars or departments.

RESOURCES (HARDWARE & SOFTWARE) TO BE USED

Hardware Requirements:

- Server with minimum 4GB RAM and 512GB storage for hosting the application.
- Minimum i3 processor.
- Workstations or personal computers for users.
- Mobile devices for accessing the mobile-friendly version.
- Client Machines: Any modern PC or laptop with a stable internet connection

Software Requirements:

- **Frontend**: HTML5, CSS3, JavaScript, Bootstrap4/5.
- **Backend**: C# , ASP.NET MVC.
- **Database**: Microsoft SQL Server (MS-SQL).
- **Version Control**: Git/GitHub.
- **Authentication**: ASP .NET Identify for User Management and Role Based Access Control (RBAC)

PROJECT SCHEDULE PLAN

Activity	Duration (Weeks)
Requirement Analysis	1
System Design	2
Frontend and Backend Setup	2
Module Development	3
Testing	1
Deployment and Maintenance	1

PROJECT TEAM

The successful execution of this project was the result of teamwork, dedication, and expertise. Below is an outline of the team members and their key responsibilities:

1. **Team Lead/Coordinator: Dr. Mirza Ghazanfar Beg**

   Responsibilities:

   - Managed overall project activities and timelines.
   - Coordinated between team members and ensured smooth communication.
   - Monitored progress to meet project goals and deliverables.

2. **Frontend Developer**: Arshil Ansari

   Responsibilities:

   - Implement user interfaces using HTML, CSS, and JavaScript.
   - Ensure responsive and cross-browser design.
   - Add interactive features for seamless user experience.
   - Optimize performance for faster loading.
   - Collaborate with back-end developers and debug issues.

3. **Backend Developer**: Syed Mohd Subhan

   Responsibilities:

   - Develop and manage server-side logic and databases.
   - Build APIs for front-end integration.
   - Ensure application performance, security, and scalability.
   - Debug, test, and fix server-side issues.
   - Collaborate with front-end developers for seamless functionality.

PROCESS DESCRIPTION

Project Overview:

Scholar Tracker is a dynamic web portal designed to assist researcher students and academic staff in managing, sharing, and accessing journal publications, research papers, and scholarly resources. The portal serves as a portfolio for researchers, helping them track and present their academic work. It also provides a platform for students to access resources provided by their respective departments or institutions. The application integrates modern web technologies including HTML, CSS, Bootstrap, JavaScript, and C# (ASP.NET MVC), with data stored in a Microsoft SQL Server database.

Development Process Overview

1. Planning & Gathering Requirements

- **Objective:** Identify user needs and define the portal's features.
- **Key Features:**
    - User registration and login (students, researchers, admins).
    - Researcher profiles to upload and share publications.
    - Access to resources for students.
    - Admin controls for managing users, resources, and submissions.
    - Search for research papers and journals.

2. System Design

- **MVC Architecture:**
    - **Model:** Manages data like users, research papers, and journals.
    - **View:** Designs the user interface (HTML, CSS, Bootstrap).
    - **Controller:** Handles user actions and connects the model and view.
- **Database Design:**
    - Tables for users, journals, and uploaded publications.
    - Track resource sharing and permissions.

3. Frontend Development

- **HTML & CSS:** Designed responsive layouts for desktop and mobile.
- **Bootstrap:** Used components (navigation bars, forms, models) for a clean, consistent look.
- **JavaScript:** Added interactivity, form validation, and dynamic content updates using AJAX.

4. Backend Development

- **ASP.NET MVC Framework:**
  - Created controllers for login, profile management, resource uploads, and searches.
  - Developed CRUD operations for managing research papers.
- **C# Logic:** Validated data, managed authentication, and handled business rules.
- **Database Connection:** Used Entity Framework and LINQ for data operations with MS-SQL.

5. User Authentication & Roles

- **User Registration/Login:** Separate roles for students, researchers, and admins.
- **Role-Based Permissions:**
  - Admins: Manage users and resources.
  - Researchers: Upload and share publications.
  - Students: Access and download resources.

6. Search Functionality

- Search for research papers, journals, and publications by keywords, authors, or topics.
- Added filters for date, department, and categories.

7. Resource Management

- **For Researchers:** Upload, update, or delete research papers and journals.
- **For Students:** Browse, download, and view metadata (title, author, description).

8. Testing

- **Unit Testing:** Checked features like login, database operations, and search.
- **Integration Testing:** Ensured smooth interaction between frontend and backend.
- **Browser Compatibility:** Tested on Chrome, Firefox, and Edge.
- **Performance Testing:** Verified the portal could handle many users at once.

9. Deployment

- **Hosting:** Deployed on Azure or on-premises with database security in place.
- **Version Control:** Used GitHub to manage code changes.

10. Post-Launch & Maintenance

- **Monitoring:** Fixed bugs and improved performance based on feedback.
- **Updates:** Added new features and security patches regularly.
- **Support:** Provided user guides and ongoing technical help.

CONTRIBUTION OF THE STUDENT

- **Requirement Analysis**: Assisted in identifying the key features and functionality needed for the project, based on user requirements.
- **Development**: Participated in designing, coding, and implementing both the front-end and back-end components of the system.
- **Testing and Debugging**: Conducted testing to identify bugs and ensure the application's reliability and performance.
- **UI/UX Design**: Contributed to designing an intuitive interface to enhance user experience.
- **Documentation**: Helped in creating technical and user documentation for the project.

CONCLUSION

Scholar Tracker provides a seamless and efficient platform for academic research management with easy to use tools for researchers and students. It's implementation of the MVC architecture ensures maintainability and scalability, while its use of modern technologies like Bootstrap and C# enhances the user experience. The portal streamlines the process of sharing and accessing scholarly publications, benefiting both research institutions and individual researchers.