PROJECT REPORT

Course Title : Information Systems Design Laboratory

Course No : CSE 3120

Project Report : Web-Based Todo Application with Secure User Authentication and Task Management

Submit Date : 12-10-2025

Submitted To,	Submitted By ,	
Md Nazirulhasan Shawon Lecturer, Department of Computer Science and Engineering	Name	: Ankon Roy, Md. Tariful Islam Jony, MD Moin Uddin Moin
	Roll	: **2107113**, **2107119**, **2107111**
Safin Ahmmed Lecturer, Department of Computer Science and Engineering	Group Year Semester	: B2 : 3rd : 1st

1. Introduction

In today's fast-paced world, people need simple and effective tools to organize their daily activities. A todo application serves as a personal assistant that helps users keep track of their tasks, set priorities, and complete their work on time.

Our project aims to develop a user-friendly web application where users can:

- Create an account and log in securely
- Add new tasks with titles and descriptions
- Mark tasks as complete or incomplete
- Update existing tasks
- Delete tasks they no longer need
- View all their tasks in one place

The application is built with security in mind, ensuring that each user's data is protected and only accessible to them. We used industry-standard technologies and followed best practices throughout the development process.

2. Objectives

Primary Objectives

1. **Develop a functional web-based todo application** that allows users to manage their tasks efficiently
2. **Implement secure user authentication** to protect user accounts and data
3. **Create a user-friendly interface** that anyone can use without technical knowledge
4. **Ensure data persistence** so users don't lose their tasks when they close the application

Secondary Objectives

1. Design and implement a proper database schema for storing user and task information
2. Follow software engineering best practices and SDLC phases systematically
3. Implement session management for maintaining user login states
4. Create comprehensive documentation including diagrams and test reports
5. Deploy the application so it can be accessed by real users

3. System Requirements

1. **User Registration**
   - Users can create accounts with email and password
   - System validates email format and password strength
   - Passwords are encrypted before storage
2. **User Login/Logout**
   - Users can log in with valid credentials
   - System maintains user sessions
   - Users can log out securely
3. **Task Management**
   - Create new tasks with title and description
   - View all tasks belonging to the logged-in user
   - Mark tasks as complete or incomplete
   - Edit existing task details
   - Delete tasks
4. **Data Security**
   - Each user can only access their own tasks
   - Passwords are hashed and never stored in plain text
   - Sessions expire after 24 hours

4. Methodology (SDLC)

We followed the **Agile SDLC methodology** with iterative sprints. Each sprint lasted one week and focused on specific features.

Sprint 1 (29 Sep - Oct 1): Planning, Desigining and Basic Setup

Phase 1: Planning

Activities:

- Defined project scope and objectives
- Identified target users and their needs
- Created initial project timeline

- Set up team communication channels
- Assigned roles and responsibilities

Deliverables:

- Project charter document
- Initial requirements list
- Sprint planning calendar

Phase 2: Requirement Analysis

Activities:

- Gathered detailed functional requirements
- Identified non-functional requirements
- Created user stories
- Prioritized features (must-have vs nice-to-have)
- Analyzed technical constraints

Deliverables:

- Software Requirements Specification (SRS) document
- User stories and acceptance criteria
- Use case diagrams

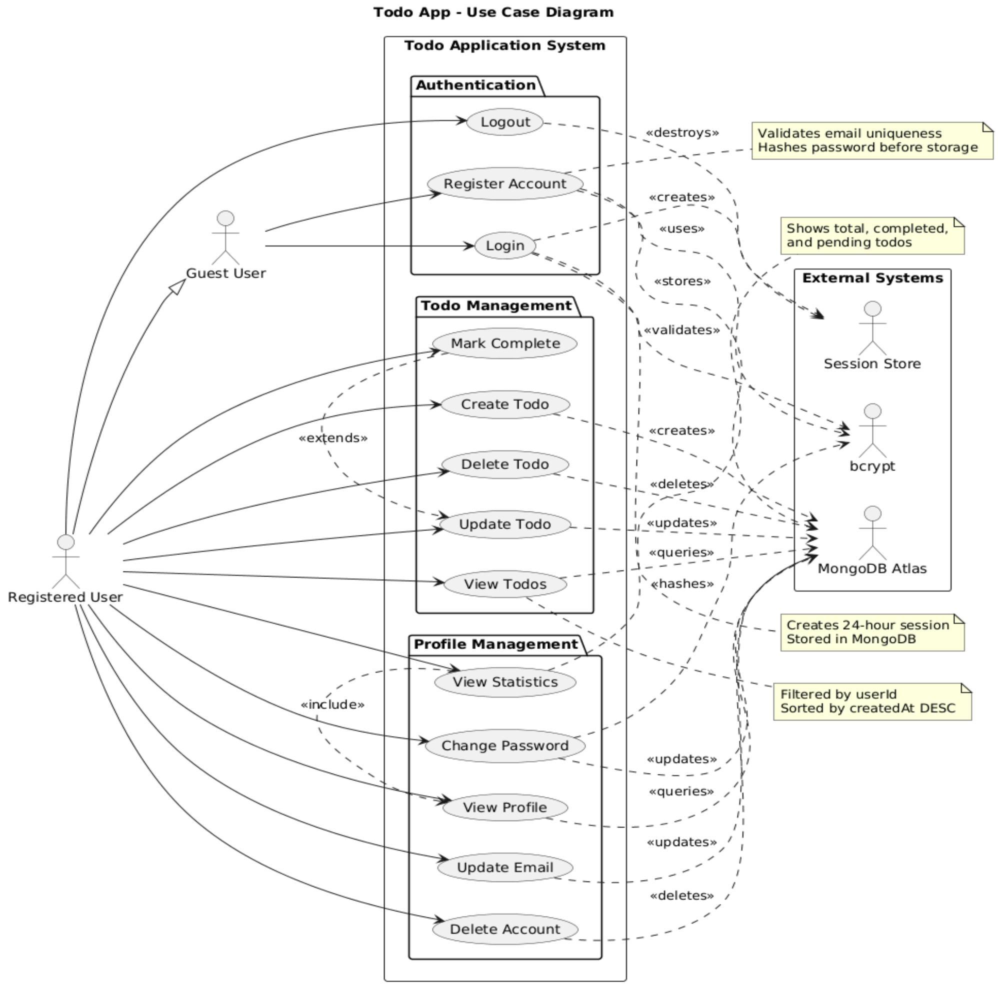

Fig 01: Use case diagram

Phase 3: System Design

Activities:

- Designed database schema
- Created system architecture
- Designed user interface mockups
- Planned API endpoints
- Created various UML diagrams

Deliverables:

- Entity-Relationship (ER) diagram

- Class diagrams
- Sequence diagrams
- Activity diagrams
- Data Flow Diagrams (DFD Level 0, 1, 2)

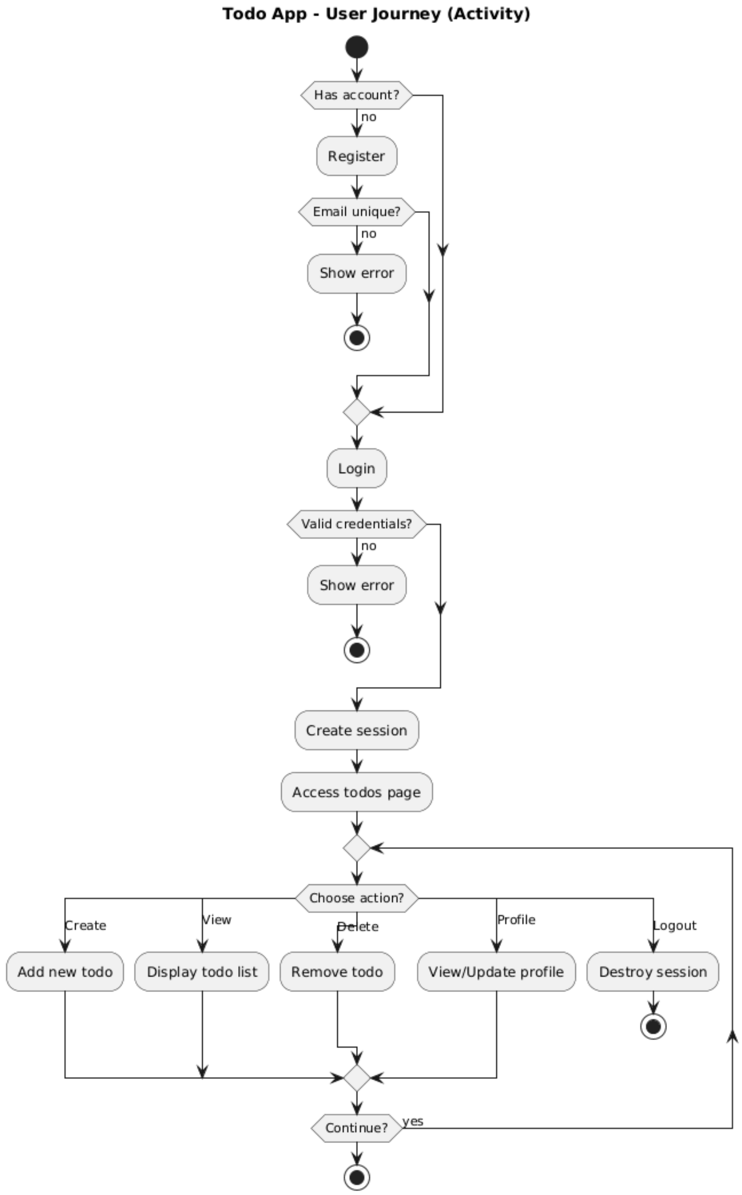

Fig 02: Activity Diagram for Todo App

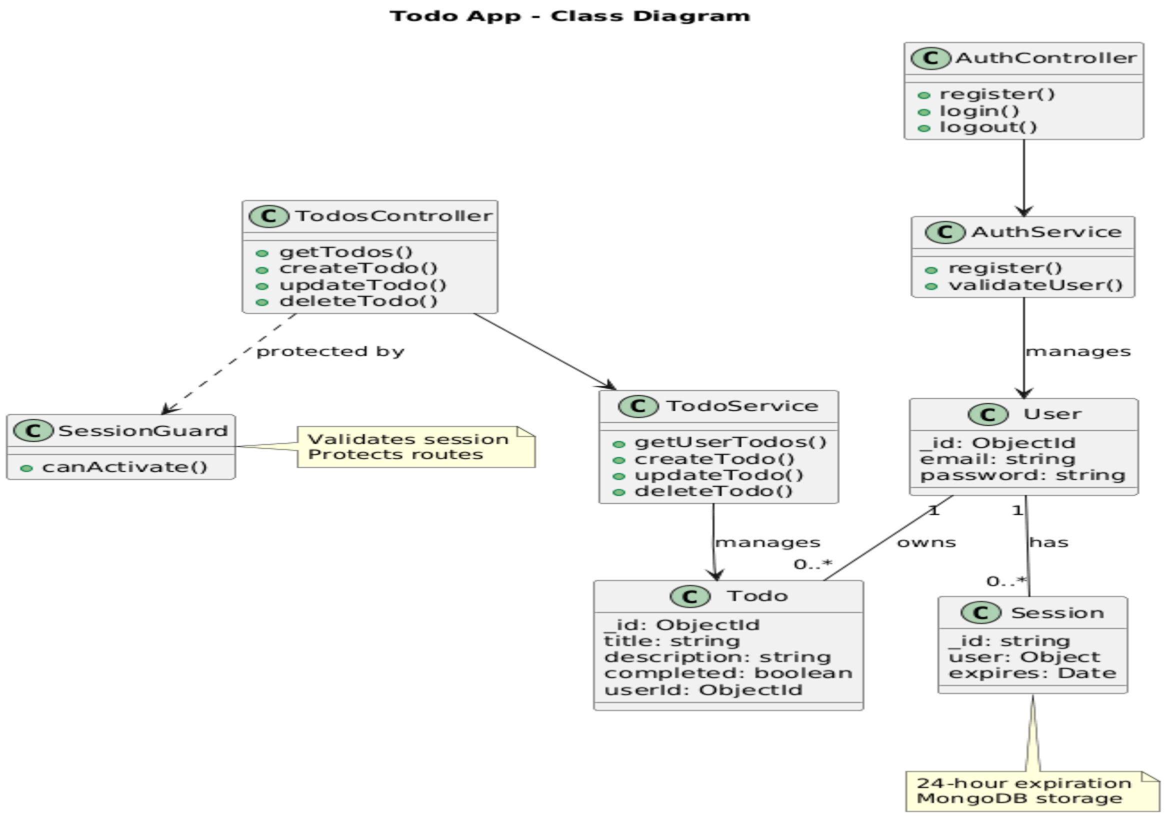

Fig-03:Class Diagram for To Do App

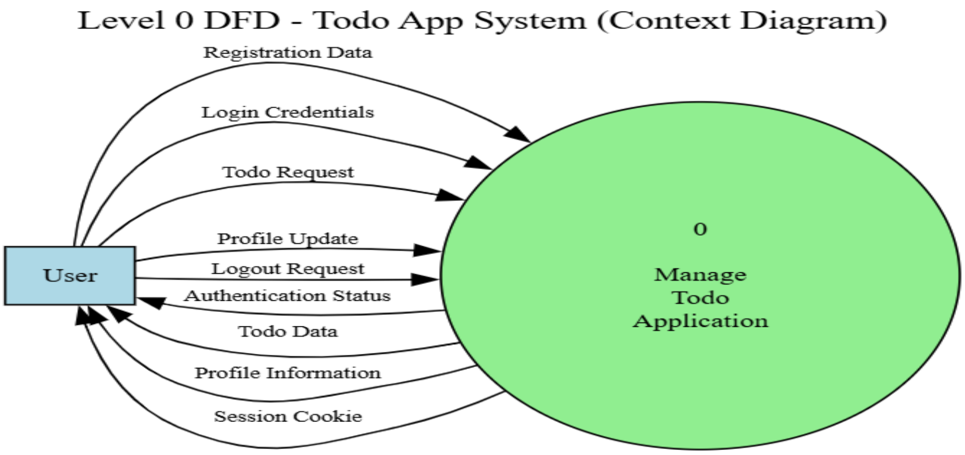

Fig-04: Level-0 DFD for Todo App

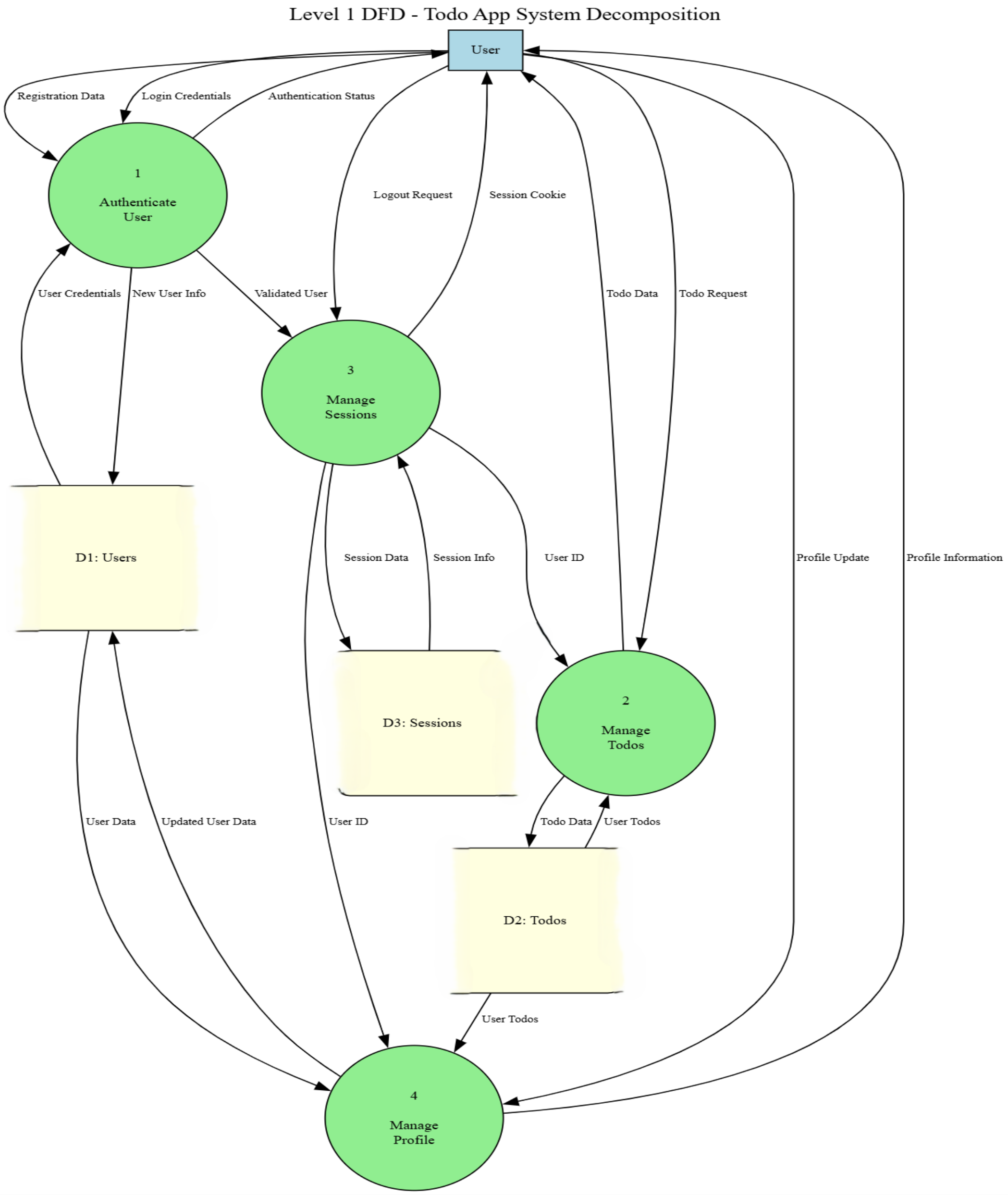

Fig-05: Level-1 DFD for Todo App

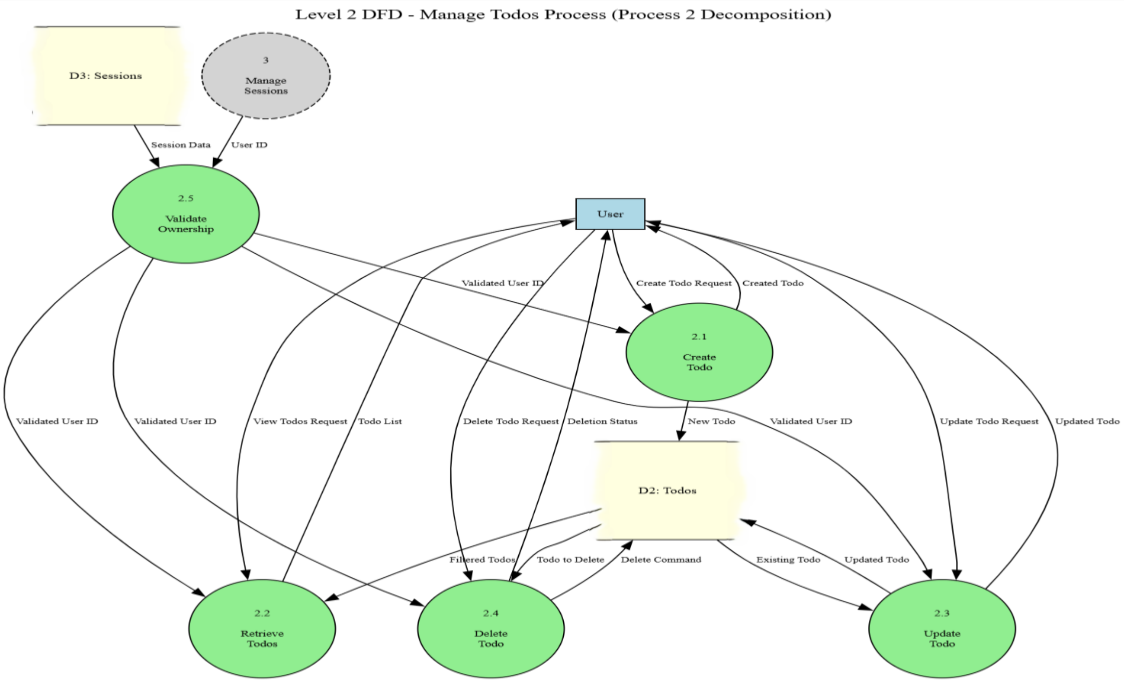

Fig-06: Level-02 DFD for Todo App

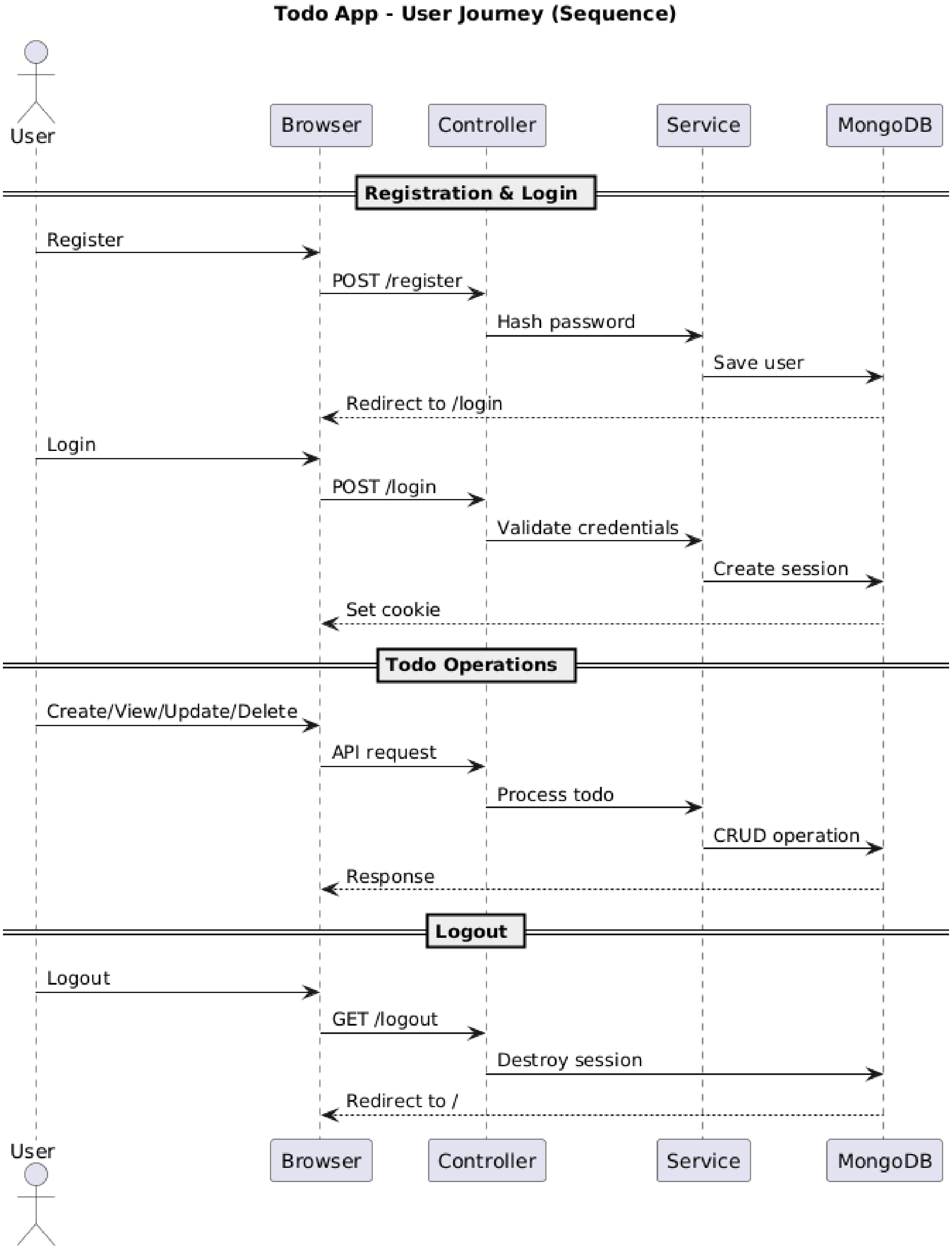

Fig-07: Sequence diagram for Todo App

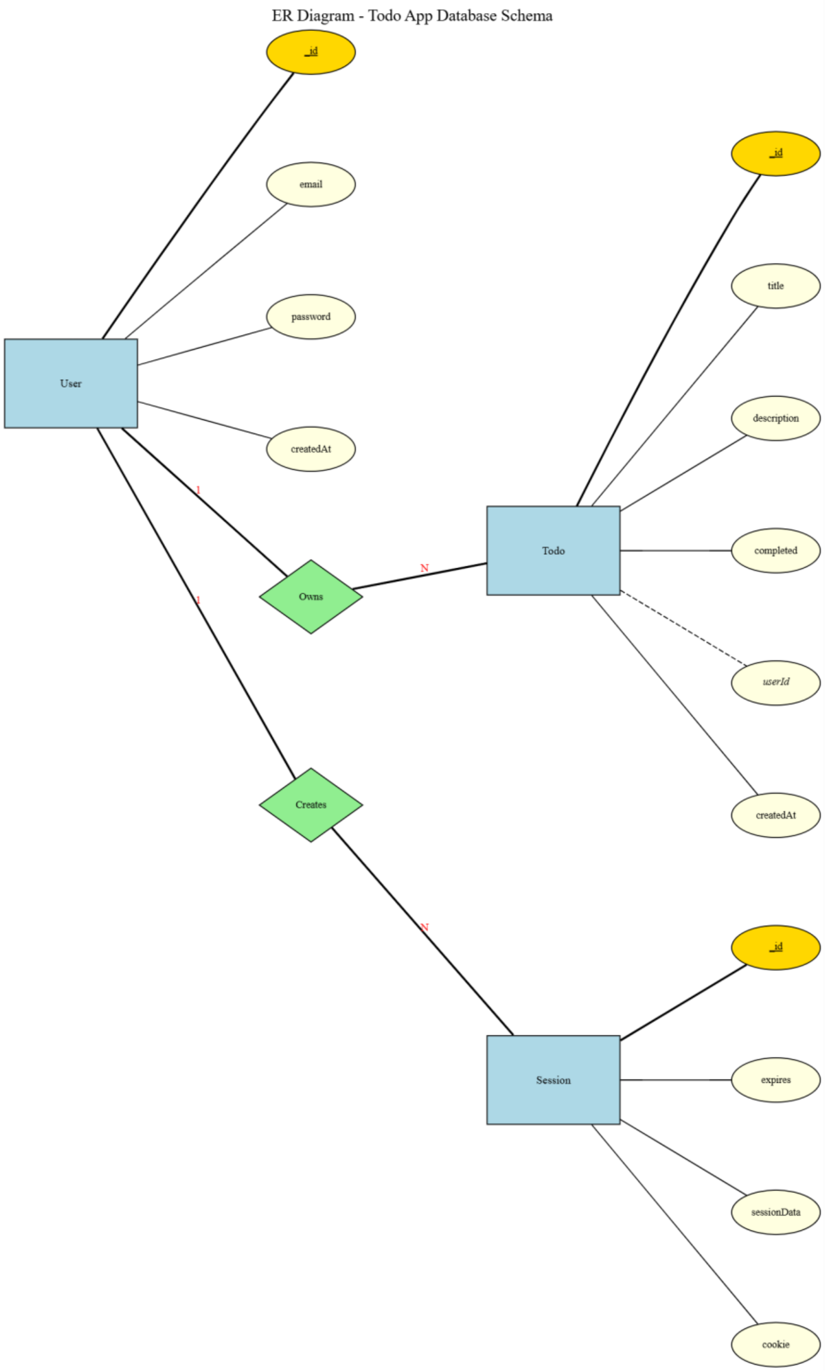

Fig-08: ER diagram for Todo App

Phase 4: Implementation (Oct 1 - Oct 13)

Sprint 2 (Oct 2 - Oct 6): Core Features

Activities:

- Set up project structure and dependencies
- Implemented user authentication system
- Created user registration and login pages
- Set up MongoDB database connection
- Implemented password hashing with bcrypt

Key Implementations:

- AuthController and AuthService for handling authentication
- User schema with encrypted password storage
- Session management with express-session
- Registration and login forms with validation

Sprint 3 (Oct 6 - Oct 8): Task Management

Activities:

- Implemented CRUD operations for todos
- Created todos display page
- Added task creation functionality
- Implemented task update and delete features
- Added task completion toggle

Key Implementations:

- TodosController and TodosService for task operations
- Todo schema with user reference
- Forms for creating and editing tasks
- Protected routes with session guards

Sprint 4 (Oct 8 - Oct 13): Enhancement and Testing

Activities:

- Improved user interface and experience
- Added error handling and validation
- Implemented security measures
- Conducted comprehensive testing
- Bug fixes and optimizations

Deliverables:

- Fully functional application
- All CRUD operations working
- Secure authentication system
- User-friendly interface

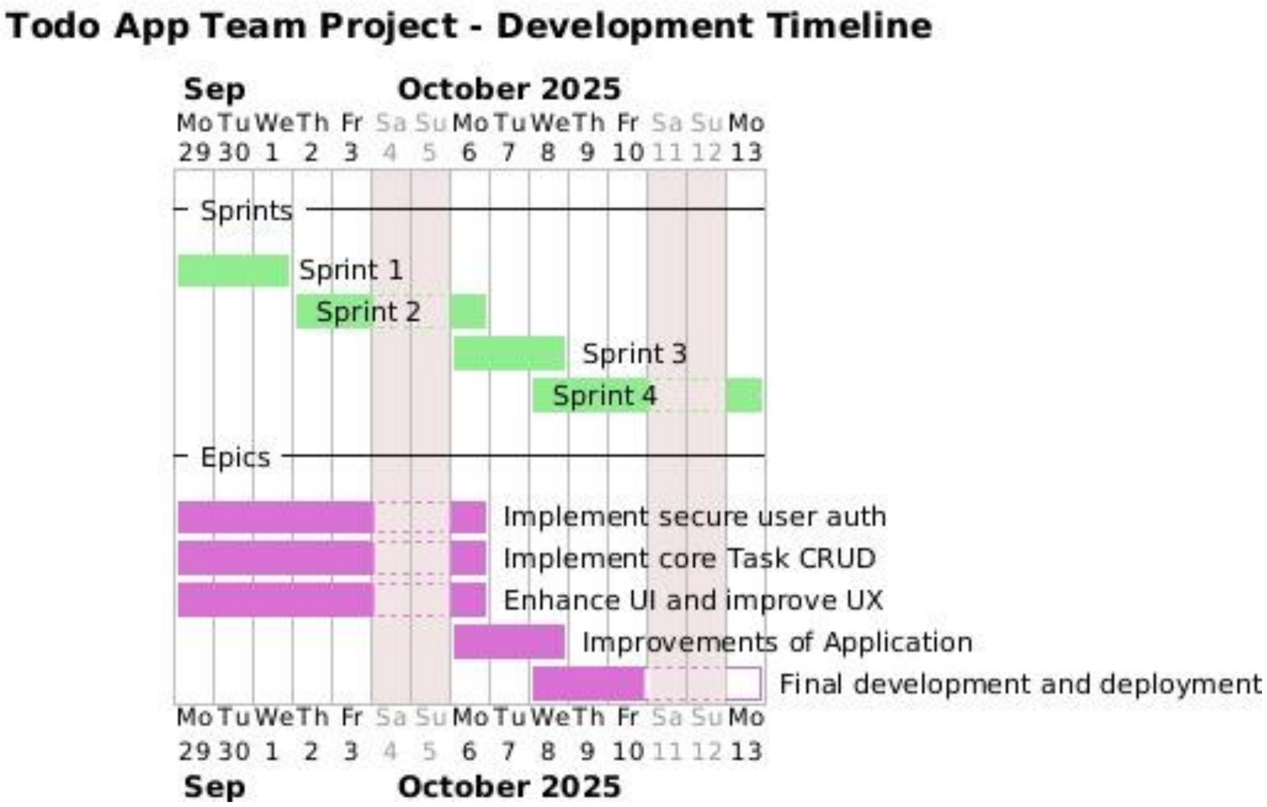

Fig -09: Gantt Chart

5.Contribution

The successful development of this Todo application was a collaborative team effort, with each member taking responsibility for specific components of the project. The contributions were distributed as follows:

Team Member Responsibilities

Ankon Roy

5. Implemented the core todo management functionality
6. Developed user profile management features
7. Created backend services for task operations

MD Moin Uddin Moin

- Designed and implemented all frontend pages
- Created responsive user interface layouts
- Integrated EJS templates with backend functionality
- Mockup and Figma Design

Tariful Islam Jony

- Developed user registration system
- Implemented login functionality
- Built secure authentication features
- Established session management

6. Discussion

Our team successfully developed a fully functional to-do application, achieving all primary objectives, including a secure authentication system with industry-standard password hashing, complete CRUD functionality for smooth task management, and a user-friendly interface with excellent performance. The project also yielded comprehensive documentation.

Throughout development, we overcame several significant challenges. Initial MongoDB connection issues taught us the importance of proper database configuration. Session management problems required correct cookie configuration, while password security implementation deepened our understanding of bcrypt hashing. We also addressed data validation gaps by implementing multi-layer checks and managed time constraints through agile sprints and effective work division.

Key lessons emerged from this experience. Technically, we learned that frameworks like NestJS, while structured, have learning curves, and that security must be integral from inception. Regarding collaboration, regular communication, code reviews, and version control proved essential. From a project management perspective, breaking work into sprints, setting realistic deadlines, and integrating documentation and testing throughout the process—rather than at the end—were crucial success factors. This project provided valuable insights into full-stack development, team dynamics, and systematic project execution that will benefit our future software engineering endeavors.

7. Conclusion

This project successfully demonstrates the development of a functional, secure, and user-friendly web-based todo application. All primary objectives were met, including the implementation of robust user authentication, complete CRUD operations, and reliable data persistence. The experience provided invaluable practical knowledge in full-stack development, modern technologies like NestJS and MongoDB, and the systematic application of the SDLC. The application not only serves as a practical tool for task management but also stands as a solid foundation for future enhancements. The skills gained in teamwork, problem-solving, and project execution will be instrumental in our future software engineering careers.