Bilkent University

Department of Computer Engineering

CS 319 Term Project

Spring 2025

Section 1

Group 1

Deliverable 5

Submission

Group Members:

Emre Algür - 22202673

Irmak İmdat - 22201570

Alper Yıldırım - 22102033

Ömer Yaslıtaş - 21902874

Melih Rıza Yıldız - 21902958

Aykhan Ahmadzada - 21903609

Instructor: Eray Tüzün

Teaching Assistant: Yahya Elnouby

1. Build Instructions

1. Prerequisites

Before you begin, ensure you have the following installed:

- **Node.js** (v18.x or later recommended): Download Node.js
- **npm** (comes with Node.js)
- **Angular CLI** (v19.2.7 or compatible):

Install globally with:

```
  npm install -g @angular/cli
```

- **Git** (for cloning the repository): Download Git

If you plan to use Docker, ensure Docker Desktop is installed, run:

```
download docker desktop
```

2. Clone the Repository

If you haven’t already, clone the project:

```
git clone https://github.com/319g1s25/TAM.git
cd TAM
git checkout working-version
```

3. Install Dependencies

Install all required Node.js packages:

```
npm install
```

This will read package.json and install all dependencies into the node_modules folder.

4. Environment Configuration

Check for a .env file in the project root. If it exists, review and update any environment variables as needed (e.g., API endpoints, database URLs, secrets).If .env is missing, create one based on .env.example or instructions in the README.

5. Database Setup

Check the db_init/ directory for SQL scripts or setup instructions. Run the necessary scripts to initialize your database. Update your .env file with the correct database connection details.

6. Running the Development Server

Start the Angular development server:

```
ng serve
```

By default, the app will be available at: http://localhost:4200/ The server will reload automatically if you change any source files.

7. Building for Production

To build the project for production deployment:

```
ng build --configuration production
```

The build artifacts will be stored in the dist/ directory.

You can serve these files using any static file server or deploy to your preferred hosting.

8. Running with Docker (Optional)

If you wish to use Docker:

```
docker compose down -v  # optional: reset volumes for a clean start
docker-compose up --build
```

Services provided with { providedIn: 'root' } are singletons by default. This means only one instance of the service exists throughout the application, which is a classic implementation of the Singleton pattern. Singleton is used to ensure a single source of truth for shared state or logic, such as authentication, notifications, or data access.

This will use the docker-compose.yaml file to build and run the application (and any services like databases, if configured).

9. Running Tests

Unit Tests

```
ng test
```

Runs unit tests using the Karma test runner.

End-to-End (E2E) Tests

```
ng e2e
```

Runs end-to-end tests (ensure you have an E2E framework configured).

2. User Manual

TAM (Teaching Assistant Management) is a web application designed to manage TA assignments, proctoring, leave requests, notifications, and performance reports for academic institutions.

Main Features

- Assign TAs to courses and manage their workloads.
- Schedule and view proctoring duties.
- Submit and manage TA leave requests.
- Receive updates and alerts.
- View and generate reports on TA performance.
- Secure login and user management.

Using the Application

Login: Navigate to http://localhost:4200/ and enter your credentials to log in.

Dashboard: After login, you'll see the dashboard with navigation to all major modules (Assignments, Proctoring, Leave Requests, etc.).

Assignments: Go to the Assignments section to view, create, or edit TA assignments.

Proctoring Calendar: Access the calendar to view upcoming proctoring duties and schedules.

Leave Requests: Submit a new leave request or view the status of existing requests.

Notifications: Check the notifications panel for important updates.

Performance Reports: Generate and view reports on TA performance and workload.

Some potential problems

Port Already in Use: If ng serve fails due to port 4200 being in use, run ng serve --port 4300 (or another port).

Dependency Issues: Run npm install again or delete node_modules and reinstall.

Database Errors: Ensure your database is running and credentials in .env are correct.

3. Workload Allocation (roughly)

Ömer: Use case diagram, side panels for different types of people, workload approve update, general about backend.

Melih Rıza: Mockup, subsystem decomposition diagram, testing and looking at edge cases, D4 and D5 reports, overview about frontend.

Aykhan: Class diagram, sequence diagram, Design goals, automatic assignment, TA proctoring swap, general about backend.

Alper: Mockup, dark mode, design goals, assignment planning and performance reports, general about frontend.

Emre: Use case descriptions, state diagram, manual filtering, add new and add new course at dashboard remove buttons, general about backend.

Irmak: General planning, use case descriptions, activity diagram, all job about databases, leave request.