**# Genix Auctions Setup Documentation**

Welcome to the Genix Auctions App setup guide. This document provides comprehensive and easy-to-follow instructions to help you set up and run the application seamlessly. As a senior full-stack developer, you will find best practices embedded in each of these steps to ensure smooth deployment and maintainability of the application.

**## Overview**

Genix Auctions App is a full-stack web application designed to facilitate online auctions. Users can register, create auction items and place bids. The application follows a modular approach and is built with Angular on the front end, Node.js with Express for the backend, and MySQL as the database.

**### Tech Stack:**

- \*\*Frontend\*\*: Angular 16 with NgRx for state management.

- \*\*Backend\*\*: Node.js with Express.js.

- \*\*Database\*\*: MySQL.

**## Prerequisites**

1. \*\*Node.js & npm\*\*: Ensure that Node.js (v14 or later) and npm (v6 or later) are installed.

2. \*\*MySQL\*\*: Install MySQL and ensure it's running locally or on a server you can access.

3. \*\*Angular CLI\*\*: Install Angular CLI globally using npm:

**## Responsiveness and Compatibility**

The Genix Auctions App is designed to be responsive and compatible across different devices and screen sizes, including:

1. Cross-Browser Compatibility: The application has been tested to work smoothly across all major browsers (Chrome, Firefox, Safari, Edge).

2. Screen Sizes: The UI is responsive and adjusts seamlessly to large, medium, and small screens, including mobile devices, ensuring a consistent user experience.

**## Database Setup**

**1. \*\*Create Database\*\***

   Open MySQL Workbench or any MySQL client, and execute the following script to create the required database schema:

   ```sql

   CREATE DATABASE auction\_app;

   USE auction\_app;

   CREATE TABLE users (

     id INT AUTO\_INCREMENT PRIMARY KEY,

     first\_name VARCHAR(50) NOT NULL,

     last\_name VARCHAR(50) NOT NULL,

     email VARCHAR(100) NOT NULL UNIQUE,

     password VARCHAR(255) NOT NULL,

     receive\_emails BOOLEAN NOT NULL DEFAULT FALSE,

     remember\_me BOOLEAN NOT NULL DEFAULT FALSE,

     created\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,

     updated\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP ON UPDATE CURRENT\_TIMESTAMP

   );

   CREATE INDEX idx\_email ON users (email);

   CREATE TABLE auctions (

     id INT AUTO\_INCREMENT PRIMARY KEY,

     title VARCHAR(100) NOT NULL,

     description TEXT NOT NULL,

     starting\_bid DECIMAL(10, 2) NOT NULL,

     current\_bid DECIMAL(10, 2) DEFAULT NULL,

     bid\_end\_date DATETIME NOT NULL,

     created\_by INT NOT NULL,

     created\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,

     updated\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP ON UPDATE CURRENT\_TIMESTAMP,

     FOREIGN KEY (created\_by) REFERENCES users(id) ON DELETE CASCADE

   );

   CREATE TABLE bids (

     id INT AUTO\_INCREMENT PRIMARY KEY,

     auction\_id INT NOT NULL,

     user\_id INT NOT NULL,

     bid\_amount DECIMAL(10, 2) NOT NULL,

     created\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,

     FOREIGN KEY (auction\_id) REFERENCES auctions(id) ON DELETE CASCADE,

     FOREIGN KEY (user\_id) REFERENCES users(id) ON DELETE CASCADE

   );

   CREATE TABLE watchlist (

     id INT AUTO\_INCREMENT PRIMARY KEY,

     user\_id INT NOT NULL,

     auction\_id INT NOT NULL,

     created\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,

     FOREIGN KEY (user\_id) REFERENCES users(id) ON DELETE CASCADE,

     FOREIGN KEY (auction\_id) REFERENCES auctions(id) ON DELETE CASCADE

   );

   CREATE INDEX idx\_auction\_end\_date ON auctions (bid\_end\_date);

   CREATE INDEX idx\_bid\_amount ON bids (bid\_amount);

   -- Insert dummy data into users table

   INSERT INTO users (first\_name, last\_name, email, password, receive\_emails, remember\_me)

   VALUES

     ('John', 'Doe', 'john.doe@example.com', 'hashed\_password\_123', TRUE, FALSE),

     ('Jane', 'Smith', 'jane.smith@example.com', 'hashed\_password\_456', FALSE, TRUE),

     ('Alice', 'Johnson', 'alice.johnson@example.com', 'hashed\_password\_789', TRUE, TRUE);

   -- Insert dummy data into auctions table

   INSERT INTO auctions (title, description, starting\_bid, current\_bid, bid\_end\_date, created\_by)

   VALUES

     ('Sony Headphones', 'Brand new noise-cancelling headphones.', 50.00, 75.00, '2024-12-31 23:59:59', 1),

     ('Apple iPhone', 'Latest model iPhone with 128GB storage.', 600.00, 650.00, '2024-11-30 23:59:59', 2),

     ('Mountain Bike', 'High-quality mountain bike suitable for all terrains.', 200.00, 250.00, '2024-10-31 23:59:59', 3);

   -- Insert dummy data into bids table

   INSERT INTO bids (auction\_id, user\_id, bid\_amount)

   VALUES

     (1, 2, 70.00),

     (1, 3, 75.00),

     (2, 1, 650.00),

     (3, 2, 240.00);

   -- Insert dummy data into watchlist table

   INSERT INTO watchlist (user\_id, auction\_id)

   VALUES

     (1, 2),

     (2, 1),

     (3, 3);

**## Developed by**

Muthu Krishna Santhosh