```
BEGIN Smart Academic Scheduler

    INITIALIZE schedule_data as empty list

    FUNCTION update_time()
        SET time_label text to current date and time
        CALL update_time every 1000 milliseconds

    FUNCTION clear_main_area()
        FOR each widget in main_area
            DESTROY widget

    FUNCTION format_date(event)
        SET text to the content of the event widget
        IF length of text is 2 or 5 THEN
            INSERT "/" at the end of text
        IF length of text is 10 THEN
            TRY
                SET day_entry to normal state
                CLEAR day_entry
                INSERT day of the week for the given date into day_entry
                SET day_entry to readonly state
            EXCEPT
                DO NOTHING

    FUNCTION format_time(event, entry)
        SET text to the content of entry
        IF text is a digit THEN
            CLEAR entry
            INSERT text padded with zeros to 2 digits followed by ":00"

    FUNCTION add_schedule()
        CALL clear_main_area()
        CALL create_schedule_form()

    FUNCTION create_schedule_form(data=None, index=None)
        DEFINE labels and keys for schedule fields
        INITIALIZE entries as empty list

        FOR each label in labels
            CREATE a Label with the label text
            CREATE an Entry for user input
            ADD Entry to entries list

        SET date_entry to bind format_date function

        IF data is provided THEN
            FOR each entry and key
                SET entry to normal state
                INSERT data into entry
                IF key is "day" THEN
                    SET entry to readonly state

        FUNCTION save()
            CREATE new_data dictionary from entries
            IF index is not None THEN
                UPDATE schedule_data at index with new_data
                SHOW message "Schedule updated successfully."
            ELSE
                APPEND new_data to schedule_data
                SHOW message "Schedule added successfully."
            CALL update_schedule()

        CREATE a Save button that calls save function
```

```
FUNCTION update_schedule()
    CALL clear_main_area()
    IF schedule_data is empty THEN
      SHOW message "No schedules available."
      RETURN
    FOR each schedule in schedule_data
      CREATE a frame for the schedule
      DISPLAY schedule information
      CREATE Edit and Delete buttons for each schedule

  FUNCTION add_schedule_for_edit(data, index)
    CALL clear_main_area()
    CALL create_schedule_form(data, index)

  FUNCTION delete_schedule(index)
    IF user confirms deletion THEN
      REMOVE schedule from schedule_data
      CALL update_schedule()

  FUNCTION view_schedule()
    CALL clear_main_area()
    IF schedule_data is empty THEN
      SHOW message "No schedules available."
      RETURN
    DISPLAY "Scheduled Entries" header
    FOR each schedule in schedule_data
      CREATE a frame for the schedule
      DISPLAY schedule information
      CREATE Edit and Delete buttons for each schedule
    CREATE an EXPORT button that calls export_schedule()

  FUNCTION export_schedule()
    IF schedule_data is empty THEN
      SHOW warning "There are no schedules to export."
      RETURN
    PROMPT user for file path to save CSV
    IF file path is not provided THEN RETURN
    TRY
      OPEN file at file path for writing
      WRITE header row to CSV
      WRITE each schedule data to CSV
      SHOW message "Schedule exported successfully."
    EXCEPT
      SHOW error message

  FUNCTION exit_app()
    IF user confirms exit THEN
      CLOSE application

  --- UI Setup ---
  INITIALIZE main application window
  SET window title and size
  CREATE header frame and label
  CALL update_time()

  CREATE body frame and button frame
  CREATE main area for displaying schedules

  FOR each button with text and command
    CREATE button and add to button frame

  DISPLAY welcome message in main area

  START main application loop

END Smart Academic Scheduler
```